



Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide, Release 10.6(x)

First Published: 2025-08-13

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 REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at <https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html>. Cisco product warranty information is available at <https://www.cisco.com/c/en/us/products/warranty-listing.html>. US Federal Communications Commission Notices are found here <https://www.cisco.com/c/en/us/products/us-fcc-notice.html>.

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 products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

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.

The documentation set for this product strives to use bias-free language. For the 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 RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2025 Cisco Systems, Inc. All rights reserved.



CONTENTS

Trademarks ?

PREFACE

Preface	xi
Audience	xi
Document Conventions	xi
Related Documentation for Cisco Nexus 9000 Series Switches	xii
Documentation Feedback	xii
Communications, services, and additional information	xii
Cisco Bug Search Tool	xiii
Documentation feedback	xiii

CHAPTER 1

New and Changed Information	1
New and Changed Information	1

CHAPTER 2

Overview	3
Licensing Requirements	3
Software Image	3
Software Compatibility	3
Spine/Leaf Topology	3
Modular Software Design	4
Serviceability	4
Switched Port Analyzer	4
Ethanalyzer	4
Smart Call Home	4
Online Diagnostics	5
Embedded Event Manager	5

- Manageability 5
 - Simple Network Management Protocol 5
 - Configuration Verification and Rollback 5
 - Role-Based Access Control 5
 - Cisco NX-OS Device Configuration Methods 5
- Programmability 6
 - Python API 6
 - Tcl 6
 - Cisco NX-API 6
 - Bash Shell 7
 - Broadcom Shell 7
- Traffic Routing, Forwarding, and Management 7
 - Ethernet Switching 7
 - IP Routing 7
 - IP Services 8
 - IP Multicast 8
- Quality of Service 9
- Network Security Features 9
- Supported Standards 9

CHAPTER 3 **Using the Cisco NX-OS Setup Utility 15**

- About the Cisco NX-OS Setup Utility 15
- Prerequisites for the Setup Utility 16
- Setting Up Your Cisco NX-OS Device 17
- Additional References for the Setup Utility 21
 - Related Documents for the Setup Utility 21

CHAPTER 4 **Using PowerOn Auto Provisioning 23**

- About PowerOn Auto Provisioning 23
 - Network Requirements for POAP 23
- POAP Process 24
 - Power-Up Phase 25
 - USB Discovery Phase 26
 - DHCP Discovery Phase 26

Script Execution Phase	27
Post-Installation Reload Phase	28
POAPv3	28
Guidelines and Limitations for POAP	29
Setting Up the Network Environment to Use POAP	31
Configuring a Switch Using POAP	31
Verifying the Device Configuration	32

CHAPTER 5**Using Network Plug and Play 33**

About Network Plug and Play	33
Guidelines and Limitations for Network Plug and Play	40
Troubleshooting Examples for Network Plug and Play	41

CHAPTER 6**Understanding the Command-Line Interface 47**

About the CLI Prompt	47
Command Modes	48
EXEC Command Mode	48
Global Configuration Command Mode	48
Interface Configuration Command Mode	49
Subinterface Configuration Command Mode	50
Saving and Restoring a Command Mode	50
Exiting a Configuration Command Mode	51
Command Mode Summary	51
Special Characters	52
Keystroke Shortcuts	53
Abbreviating Commands	56
Completing a Partial Command Name	56
Identifying Your Location in the Command Hierarchy	57
Using the no Form of a Command	57
Configuring CLI Variables	58
About CLI Variables	58
Configuring CLI Session-Only Variables	59
Configuring Persistent CLI Variables	59
Command Aliases	60

About Command Aliases	60
Defining Command Aliases	61
Configuring Command Aliases for a User Session	62
Command Scripts	62
Running a Command Script	63
Echoing Information to the Terminal	63
Delaying Command Action	64
Context-Sensitive Help	64
Understanding Regular Expressions	66
Special Characters	66
Multiple-Character Patterns	66
Anchoring	67
Searching and Filtering show Command Output	67
Filtering and Searching Keywords	68
diff Utility	69
grep and egrep Utilities	70
less Utility	70
Mini AWK Utility	71
sed Utility	71
sort Utility	71
Searching and Filtering from the --More-- Prompt	72
Using the Command History	73
Recalling a Command	73
Controlling CLI History Recall	73
Configuring the CLI Edit Mode	73
Displaying the Command History	74
Enabling or Disabling the CLI Confirmation Prompts	75
Setting CLI Display Colors	75
Sending Commands to Modules	76
Sending Command Output in Email	77
BIOS Loader Prompt	78
Examples Using the CLI	79
Using the System-Defined Timestamp Variable	79
Using CLI Session Variables	79

Defining Command Aliases	80
Running a Command Script	80
Sending Command Output in Email	81

CHAPTER 7	Configuring Terminal Settings and Sessions	83
	About Terminal Settings and Sessions	83
	Terminal Session Settings	83
	Console Port	83
	Virtual Terminals	84
	Default Settings for File System Parameters	84
	Configuring the Console Port	84
	Configuring Virtual Terminals	86
	Configuring the Inactive Session Timeout	86
	Configuring the Session Limit	87
	Clearing Terminal Sessions	88
	Displaying Terminal and Session Information	88

CHAPTER 8	Basic Device Management	91
	About Basic Device Management	91
	Device Hostname	91
	Message-of-the-Day Banner	91
	Device Clock	91
	Clock Manager	92
	Time Zone and Summer Time (Daylight Saving Time)	92
	User Sessions	92
	Default Settings for Basic Device Parameters	92
	Changing the Device Hostname	92
	Configuring the MOTD Banner	93
	Configuring the Time Zone	95
	Configuring Summer Time (Daylight Saving Time)	96
	Manually Setting the Device Clock	97
	Setting the Clock Manager	97
	Managing Users	98
	Displaying Information about the User Sessions	98

Sending a Message to Users	99
Verifying the Device Configuration	99

CHAPTER 9

Using the Device File Systems, Directories, and Files	101
About the Device File Systems, Directories, and Files	101
File Systems	101
Directories	102
Files	102
Default Settings for File System Parameters	103
Configuring the FTP, HTTP, or TFTP Source Interface	103
Working with Directories	104
Identifying the Current Directory	104
Changing the Current Directory	104
Creating a Directory	105
Displaying Directory Contents	105
Deleting a Directory	106
Accessing Directories on the Standby Supervisor Module	106
Working with Files	106
Moving Files	107
Copying Files	107
Deleting Files	108
Displaying File Contents	109
Displaying File Checksums	109
Compressing and Uncompressing Files	110
Displaying the Last Lines in a File	110
Redirecting show Command Output to a File	111
Finding Files	111
Formatting the Bootflash	112
Working with Archive Files	113
Creating an Archive File	113
Appending Files to an Archive File	114
Extracting Files from an Archive File	115
Displaying the Filenames in an Archive File	115
Examples of Using the File System	116

Accessing Directories on Standby Supervisor Modules	116
Moving Files	116
Copying Files	116
Deleting a Directory	117
Displaying File Contents	117
Displaying File Checksums	118
Compressing and Uncompressing Files	118
Redirecting show Command Output	118
Finding Files	119

CHAPTER 10

Working with Configuration Files	121
About Configuration Files	121
Types of Configuration Files	121
Guidelines and Limitations for Configuration Files	122
Managing Configuration Files	122
Saving the Running Configuration to the Startup Configuration	122
Copying a Configuration File to a Remote Server	122
Downloading the Running Configuration From a Remote Server	123
Downloading the Startup Configuration From a Remote Server	124
Copying Configuration Files to an External Flash Memory Device	126
Copying the Running Configuration from an External Flash Memory Device	127
Copying the Startup Configuration From an External Flash Memory Device	128
Copying Configuration Files to an Internal File System	128
Rolling Back to a Previous Configuration	129
Removing the Configuration for a Missing Module	130
Erasing a Configuration	131
Clearing Inactive Configurations	131
Verifying the Device Configuration	132
Examples of Working with Configuration Files	133
Copying Configuration Files	133
Backing Up Configuration Files	133
Rolling Back to a Previous Configuration	133



Preface

This preface includes the following sections:

- [Audience, on page xi](#)
- [Document Conventions, on page xi](#)
- [Related Documentation for Cisco Nexus 9000 Series Switches, on page xii](#)
- [Documentation Feedback, on page xii](#)
- [Communications, services, and additional information, on page xii](#)

Audience

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

Document Conventions

Command descriptions use the following conventions:

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

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

Examples use the following conventions:

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

Related Documentation for Cisco Nexus 9000 Series Switches

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

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

Documentation Feedback

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

Communications, services, and additional information

- To receive timely, relevant information from Cisco, sign up at [Cisco Profile Manager](#).
- To get the business impact you're looking for with the technologies that matter, visit [Cisco Services](#).
- To submit a service request, visit [Cisco Support](#).
- To discover and browse secure, validated enterprise-class apps, products, solutions, and services, visit [Cisco DevNet](#).
- 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 gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

Documentation feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.



CHAPTER 1

New and Changed Information

- [New and Changed Information](#), on page 1

New and Changed Information

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
POAP support on N9300 Smart Switches	Added support for POAP over out-of-band (management) interface on N9164E-NS4-O switch	10.6(2n)F	Guidelines and Limitations for POAP , on page 29



CHAPTER 2

Overview

This chapter contains these sections:

- [Licensing Requirements, on page 3](#)
- [Software Image, on page 3](#)
- [Software Compatibility, on page 3](#)
- [Serviceability, on page 4](#)
- [Manageability, on page 5](#)
- [Programmability, on page 6](#)
- [Traffic Routing, Forwarding, and Management, on page 7](#)
- [Quality of Service, on page 9](#)
- [Network Security Features, on page 9](#)
- [Supported Standards, on page 9](#)

Licensing Requirements

See the [Cisco NX-OS Licensing Guide](#) and [Cisco NX-OS Licensing Options Guide](#) for Cisco NX-OS licensing recommendations and instructions to obtain and apply licenses.

Software Image

The Cisco NX-OS software consists of one NXOS software image.

Software Compatibility

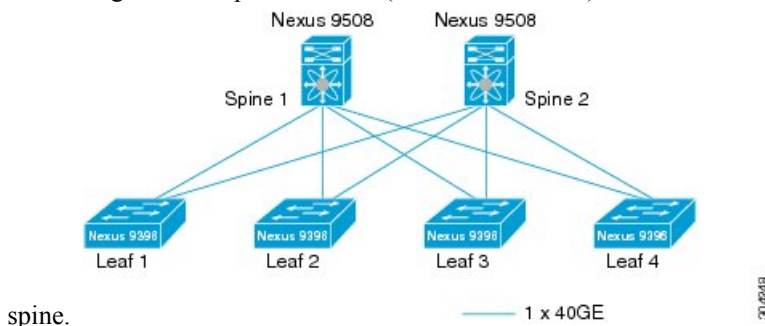
The Cisco NX-OS software interoperates with Cisco products that run any variant of the Cisco IOS software. The Cisco NX-OS software also interoperates with any networking operating system that conforms to the IEEE and RFC compliance standards.

Spine/Leaf Topology

The Cisco Nexus 9000 Series switches support a two-tier spine/leaf topology.

Figure 1: Spine/Leaf Topology

This figure shows an example of a spine/leaf topology with four leaf switches (Cisco Nexus 9396 or 93128) connecting into two spine switches (Cisco Nexus 9508) and two 40G Ethernet uplinks from each leaf to each



spine.

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.

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 9000 Series NX-OS System Management Configuration Guide*.

Ethalyzer

Ethalyzer is a Cisco NX-OS protocol analyzer tool based on the Wireshark (formerly Ethereal) open source code. Ethalyzer is a command-line version of Wireshark for capturing and decoding packets. You can use Ethalyzer to troubleshoot your network and analyze the control-plane traffic. For more information about Ethalyzer, see the *Cisco Nexus 9000 Series NX-OS Troubleshooting Guide*.

Smart 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 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 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 Smart Call Home, see the *Cisco Nexus 9000 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 9000 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 9000 Series NX-OS System Management Configuration Guide*.

Manageability

This section describes the manageability features for the Cisco Nexus 9000 Series switches.

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 9000 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 9000 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 9000 Series NX-OS Security Configuration Guide*.

Cisco NX-OS Device Configuration Methods

You can use these methods to configure Cisco NX-OS devices:

- The CLI from a Secure Shell (SSH) session, a Telnet session, or the console port. SSH provides a secure connection to the device. The CLI configuration guides are organized by feature. For more information, see the Cisco NX-OS configuration guides. For more information about SSH and Telnet, see the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.
- The XML management interface, which is a programmatic method based on the NETCONF protocol that complements the CLI. For more information, see the *Cisco NX-OS XML Interface User Guide*.
- The Cisco Data Center Network Management (DCNM) client, which runs on your local PC and uses web services on the Cisco DCNM server. The Cisco DCNM server configures the device over the XML management interface. For more information about the Cisco DCNM client, see the *Cisco DCNM Fundamentals Guide*.

Programmability

This section describes the programmability features for the Cisco Nexus 9000 Series switches.

Python API

Python is an easy-to-learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python website: <http://www.python.org/>. The Python scripting capability gives programmatic access to the CLI to perform various tasks and Power-On Auto Provisioning (POAP) or Embedded Event Manager (EEM) actions. For more information about the Python API and Python scripting, see the *Cisco Nexus 9000 Series NX-OS Programmability Guide*.

Tcl

Tool Command Language (Tcl) is a scripting language. 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.

Cisco NX-API

The Cisco NX-API provides web-based programmatic access to the Cisco Nexus 9000 Series switches. This support is delivered through the NX-API open-source web server. The Cisco NX-API exposes the complete configuration and management capabilities of the command-line interface (CLI) through web-based APIs. You can configure the switch to publish the output of the API calls in either XML or JSON format. For more information about the Cisco NX-API, see the *Cisco Nexus 9000 Series NX-OS Programmability Guide*.



Note NX-API performs authentication through a programmable authentication module (PAM) on the switch. Use cookies to reduce the number of PAM authentications and thus reduce the load on PAM.

Bash Shell

The Cisco Nexus 9000 Series switches support direct Linux shell access. With Linux shell support, you can access the Linux system on the switch in order to use Linux commands and manage the underlying system. For more information about Bash shell support, see the *Cisco Nexus 9000 Series NX-OS Programmability Guide*.

Broadcom Shell

The Cisco Nexus 9000 Series switch front-panel and fabric module line cards contain several Broadcom ASICs. You can use the CLI to access the command-line shell (bcm shell) for these ASICs. The benefit of using this method to access the bcm shell is that you can use Cisco NX-OS command extensions such as **pipe include** and **redirect output to file** to manage the output. In addition, the activity is recorded in the system accounting log for audit purposes, unlike commands entered directly from the bcm shell, which are not recorded in the accounting log. For more information about Broadcom shell support, see the *Cisco Nexus 9000 Series NX-OS Programmability Guide*.



Caution Use Broadcom shell commands with caution and only under the direct supervision or request of Cisco Support personnel.

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
- IEEE 802.3ad link aggregation
- Unidirectional Link Detection (UDLD) in aggressive and standard modes

For more information, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide* and the *Cisco Nexus 9000 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 (IPv4 and IPv6)
- Border Gateway Protocol (BGP) (IPv4 and IPv6)
- Enhanced Interior Gateway Routing Protocol (EIGRP) (IPv4 only)
- Routing Information Protocol Version 2 (RIPv2) (IPv4 only)

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, and loopback interfaces.

For more information, see the *Cisco Nexus 9000 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 Router Protocol (HSRP)
- 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 9000 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)
- PIM sparse mode (Any-Source Multicast [ASM] for IPv4)
- Anycast rendezvous point (Anycast-RP)
- Multicast NSF for IPv4
- 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 Source Discovery Protocol (MSDP) (for IPv4)



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

For more information, see the *Cisco Nexus 9000 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 9000 Series NX-OS Quality of Service Configuration Guide*.

Network Security Features

The Cisco NX-OS software includes the following security features:

- Control Plane Policing (CoPP)
- Message-digest algorithm 5 (MD5) routing protocol authentication
- Authentication, authorization, and accounting (AAA)
- RADIUS and TACACS+
- SSH Protocol Version 2
- SNMPv3
- 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)

For more information, see the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Supported Standards

This table lists the IEEE compliance standards.

Table 2: IEEE Compliance Standards

Standard	Description
802.1D	MAC Bridges
802.1p	Class of Service Tagging for Ethernet frames
802.1Q	VLAN Tagging
802.1s	Multiple Spanning Tree Protocol

Standard	Description
802.1w	Rapid Spanning Tree Protocol
802.3ab	1000Base-T (10/100/1000 Ethernet over copper)
802.3ad	Link aggregation with LACP
802.3ae	10-Gigabit Ethernet

This table lists the RFC compliance standards. For information on each RFC, see www.ietf.org.

Table 3: RFC Compliance Standards

Standard	Description
BGP	
RFC 1997	<i>BGP Communities Attribute</i>
RFC 2385	<i>Protection of BGP Sessions via the TCP MD5 Signature Option</i>
RFC 2439	<i>BGP Route flap damping</i>
RFC 2519	<i>A Framework for Inter-Domain Route Aggregation</i>
RFC 2545	<i>Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 2918	<i>Route Refresh Capability for BGP-4</i>
RFC 3065	<i>Autonomous System Confederations for BGP</i>
RFC 3392	<i>Capabilities Advertisement with BGP-4</i>
RFC 4271	<i>BGP version 4</i>
RFC 4273	<i>BGP4 MIB - Definitions of Managed Objects for BGP-4</i>
RFC 4456	<i>BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)</i>
RFC 4486	<i>Subcodes for BGP cease notification message</i>

Standard	Description
RFC 4724	<i>Graceful Restart Mechanism for BGP</i>
RFC 4893	<i>BGP Support for Four-octet AS Number Space</i>
RFC 5004	<i>Avoid BGP Best Path Transitions from One External to Another</i>
RFC 5396	<i>Textual Representation of Autonomous System (AS) Numbers</i> Note RFC 5396 is partially supported. The asplain and asdot notations are supported, but the asdot+ notation is not.
RFC 5549	<i>Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop</i>
RFC 5668	<i>4-Octet AS Specific BGP Extended Community</i>
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)
IP Multicast	
RFC 2236	<i>Internet Group Management Protocol, Version 2</i>
RFC 3376	<i>Internet Group Management Protocol, Version 3</i>
RFC 3446	<i>Anycast Rendezvous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)</i>
RFC 3569	<i>An Overview of Source-Specific Multicast (SSM)</i>

Standard	Description
RFC 3618	<i>Multicast Source Discovery Protocol (MSDP)</i>
RFC 4601	<i>Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)</i>
RFC 4607	<i>Source-Specific Multicast for IP</i>
RFC 4610	<i>Anycast-RP Using Protocol Independent Multicast (PIM)</i>
RFC 6187	<i>X.509v3 Certificates for Secure Shell Authentication</i>
RFC 9465	<i>PIM Null-Register Packing</i>
ietf-draft	Mtrace server functionality, to process mtrace-requests, draft-ietf-idmr-traceroute-ipm-07.txt
IP Services	
RFC 768	<i>UDP</i>
RFC 783	<i>TFTP</i>
RFC 791	<i>IP</i>
RFC 792	<i>ICMP</i>
RFC 793	<i>TCP</i>
RFC 826	<i>ARP</i>
RFC 854	<i>Telnet</i>
RFC 959	<i>FTP</i>
RFC 1027	<i>Proxy ARP</i>
RFC 8573	<i>NTP security is enhanced with the AES128CMAC authentication mechanism</i>
RFC 1305	<i>NTP v3</i>
RFC 1519	<i>CIDR</i>
RFC 1542	<i>BootP relay</i>
RFC 1591	<i>DNS client</i>
RFC 1812	<i>IPv4 routers</i>

Standard	Description
RFC 2131	<i>DHCP Helper</i>
RFC 2338	<i>VRRP</i>
IS-IS	
RFC 1142 (OSI 10589)	<i>OSI 10589 Intermediate system to intermediate system intra-domain routing exchange protocol</i>
RFC 1195	<i>Use of OSI IS-IS for routing in TCP/IP and dual environment</i>
RFC 2763	<i>Dynamic Hostname Exchange Mechanism for IS-IS</i>
RFC 2966	<i>Domain-wide Prefix Distribution with Two-Level IS-IS</i>
RFC 2973	<i>IS-IS Mesh Groups</i>
RFC 3277	<i>IS-IS Transient Blackhole Avoidance</i>
RFC 3373	<i>Three-Way Handshake for IS-IS Point-to-Point Adjacencies</i>
RFC 3567	<i>IS-IS Cryptographic Authentication</i>
RFC 3847	<i>Restart Signaling for IS-IS</i>
ietf-draft	Internet Draft Point-to-point operation over LAN in link-state routing protocols (draft-ietf-isis-igp-p2p-over-lan-06.txt)
OSPF	
RFC 2328	<i>OSPF Version 2</i>
RFC 2370	<i>OSPF Opaque LSA Option</i>
RFC 2740	<i>OSPF for IPv6 (OSPF version 3)</i>
RFC 3101	<i>OSPF Not-So-Stubby-Area (NSSA) Option</i>
RFC 3137	<i>OSPF Stub Router Advertisement</i>
RFC 3509	<i>Alternative Implementations of OSPF Area Border Routers</i>
RFC 3623	<i>Graceful OSPF Restart</i>

Standard	Description
RFC 4750	<i>OSPF Version 2 MIB</i>
RIP	
RFC 1724	<i>RIPv2 MIB extension</i>
RFC 2082	<i>RIPv2 MD5 Authentication</i>
RFC 2453	<i>RIP Version 2</i>
RFC 2819	<i>Remote Network Monitoring Management Information Base</i>
RFC 3164	<i>The BSD syslog Protocol</i>
RFC 3411 and RFC 3418	<i>An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks</i>



CHAPTER 3

Using the Cisco NX-OS Setup Utility

This chapter contains these sections:

- [About the Cisco NX-OS Setup Utility, on page 15](#)
- [Prerequisites for the Setup Utility, on page 16](#)
- [Setting Up Your Cisco NX-OS Device, on page 17](#)
- [Additional References for the Setup Utility, on page 21](#)

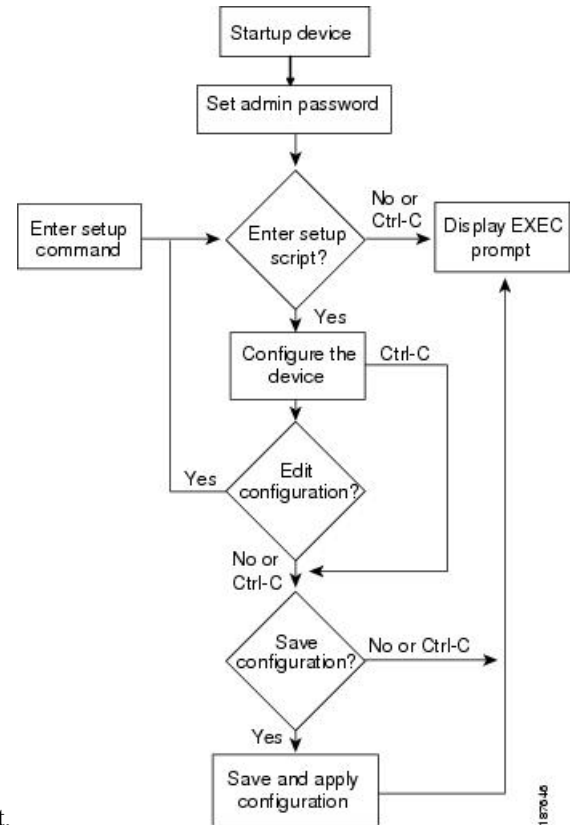
About the Cisco NX-OS Setup Utility

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.

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.

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 ----
```

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 Nexus 9000 Family devices promptly with your supplier. Failure to register may affect response times for initial service calls. Nexus 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) [network-operator]: *default_user_role*

For information on the default user roles, see the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

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 9000 Series NX-OS System Management Configuration Guide*.

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 9000 Series NX-OS Unicast Routing Configuration Guide*.

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 9000 Series NX-OS Security Configuration Guide*.

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 9000 Series NX-OS System Management Configuration Guide*.

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 9000 Series NX-OS Interfaces Configuration Guide*.

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 9000 Series NX-OS Security Configuration Guide*.

Example:

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

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

Step 21 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 22 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 nx-os image 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 nx-os image 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	<i>Cisco NX-OS Licensing Guide</i>
SSH and Telnet	<i>Cisco Nexus 9000 Series NX-OS Security Configuration Guide</i>
User roles	<i>Cisco Nexus 9000 Series NX-OS Security Configuration Guide</i>
IPv4 and IPv6	<i>Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide</i>

Related Topic	Document Title
SNMP and NTP	<i>Cisco Nexus 9000 Series NX-OS System Management Configuration Guide</i>



CHAPTER 4

Using PowerOn Auto Provisioning

This chapter contains the following sections:

- [About PowerOn Auto Provisioning, on page 23](#)
- [POAPv3, on page 28](#)
- [Guidelines and Limitations for POAP, on page 29](#)
- [Setting Up the Network Environment to Use POAP, on page 31](#)
- [Configuring a Switch Using POAP, on page 31](#)
- [Verifying the Device Configuration, on page 32](#)

About PowerOn Auto Provisioning

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

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



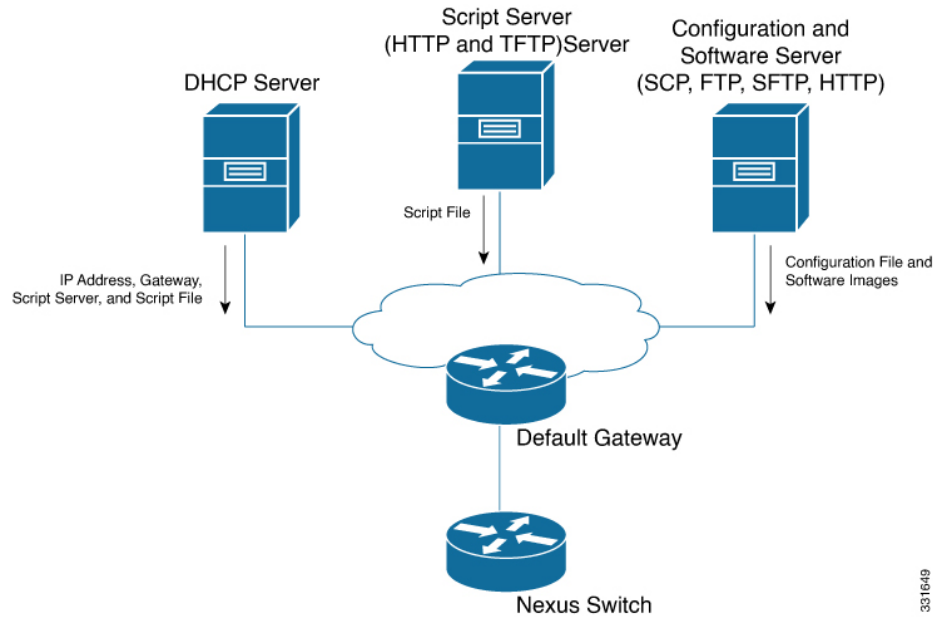
Note The DHCP information is used only during the POAP process.

Network Requirements for POAP

POAP requires the following network infrastructure:

- A DHCP server to bootstrap the interface IP address, gateway address, and Domain Name System (DNS) 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.
- If you use USB, then no DHCP server or TFTP server are required for POAP.

Figure 3: POAP Network Infrastructure



331649

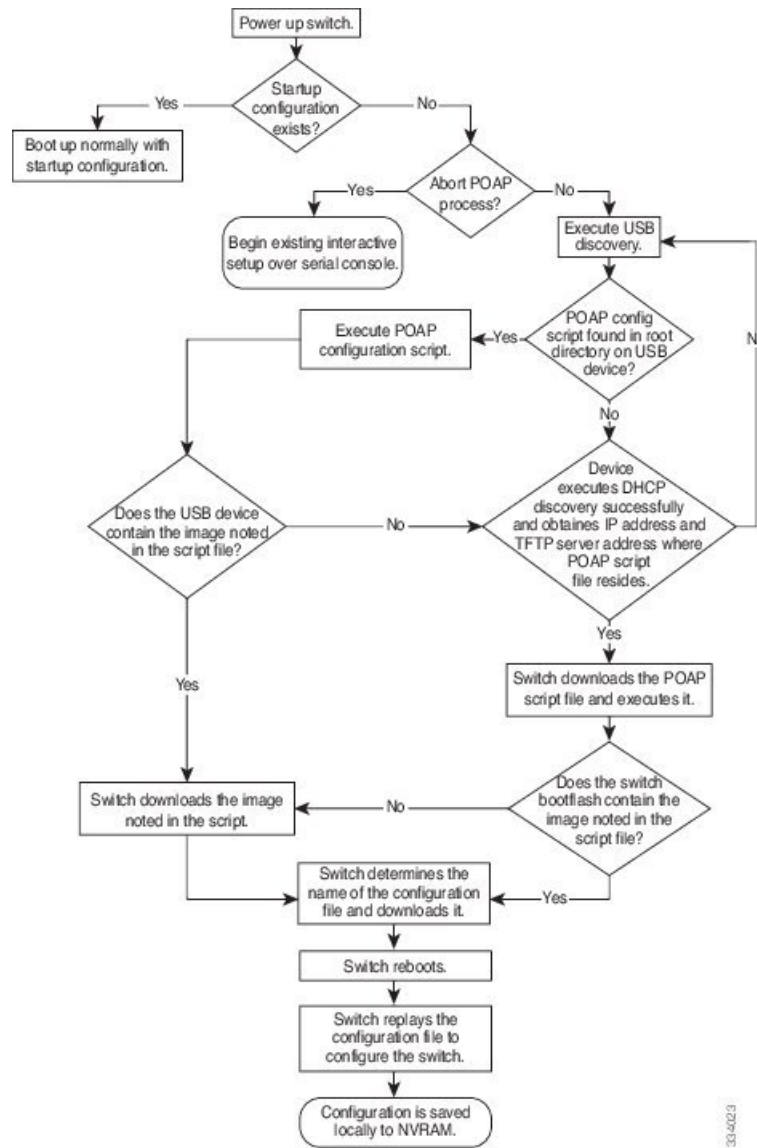
POAP Process

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 4: POAP Process



Power-Up Phase

When you powerup the device 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.

USB Discovery Phase

When POAP starts, the process searches the root directory of all accessible USB devices for the POAP script file (the Python script file, `poap_script.py`), configuration files, and system and kickstart images.

If the script file is found on a USB device, POAP begins running the script. If the script file is not found on the USB device, POAP executes DHCP discovery. (When failures occur, the POAP process alternates between USB discovery and DHCP discovery, until POAP succeeds or you manually abort the POAP process.)

If the software image and switch configuration files specified in the configuration script are present, POAP uses those files to install the software and configure the switch. If the software image and switch configuration files are not on the USB device, POAP does some cleanup and starts DHCP phase from the beginning.

DHCP Discovery Phase

The switch sends out DHCP discover messages on the front-panel interfaces or the MGMT interface that solicit 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.

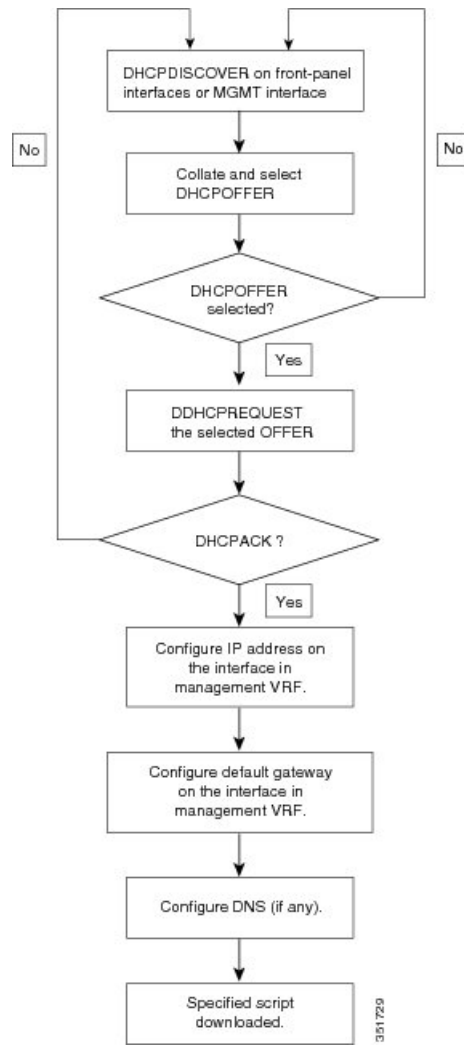
The DHCP discover message also solicits the following options from the DHCP server:

- 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, the one arriving first is honored and the POAP process moves to next stage. 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 5: DHCP Discovery Process



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.

POAPv3

PowerOn Auto Provisioning version 3 (POAPv3) is introduced in Cisco NX-OS Release 9.3(5). With this feature you can install license, RPM, and certificate through POAP.

Perform the following steps to install license or RPM or certificate through POAP.

1. Create a folder on the POAP server with serial number of the box as the name.
2. Create .yaml or .yml file with files to be installed. Make sure the file name is in <serial-number>.yaml or <serial-number>.yml format.
3. Create MD5 checksum for the .yaml or .yml file.
4. Make sure the format of the .yaml file should be similar to the below format:

```
Version : 1

Target-image : nxos.9.3.4.bin

Description : Yaml for box XYZ12345 poap provisioning. N9k Leaf mode box

License : [license1.lic, XYZ12345/license2.lic, folder1/license3.lic]

RPM :

- rpml.rpm

- patches/reload/rpm2-reload.rpm

- rpm3.rpm

Certificate : [ssh1.pub, XYZ12345/ssh2key.pub]

Trustpoint :

CA1 :

cert_1.p12 : password1 (priv_key_passphrase)

XYZ12345/CA1/cert_2.pfx : password2

CA2 :

CA2/XYZ12345/cert_3.p12 : password3
```

5. Note that the yaml keywords must match the format shown in above example.
6. Place all files in appropriate path.
7. Update the POAP script with install_path variable as the path where folder with the serial number as name is placed.

The following list provides the guidelines and limitations related to POAPv3:

- YAML is a human friendly data serialization standard for all programming languages. YAML stands for YAML Ain't Markup Language, and this file format technology is used in documents. These documents are saved in plain text format and are appended with the .yml extension. YAML is the file format and .yml is the file extension.
- YAML is a superset of JSON and the YAML parser understands JSON. YAML file formats are used for configuration management because it is easy to read and comments are useful.
- The Target_image mentioned in yaml should be kept only in the target_system_image path mentioned within POAP script. Relative path is not supported for the Target_image in yaml file.
- Both .yaml and .yml extensions are supported. You have an option to choose to use any of these extensions. If you don't choose any option, the <serial>.yaml extension will be tried first and if it fails the <serial>.yml is considered.
- The MD5 files of yaml/yml is required similar to the configuration file. But if the disable_md5 is 'True' then the MD5 files of yaml/yml are not required.
- Although 'install_path' is set in the POAP script file if no yaml file for device is found, then POAP workflow will proceed with the legacy path, i.e., without any installation of RPMs, licenses and certificates.
- Install reset is highly preferred over write erase if PoAP with RPM installation is done in scenarios apart from Day-0.
- ISSU is the new default for moving to new image via PoAP. Note that you need to use "use_nxos_boot": True, if legacy boot nxos <> is required.
- The Filetype checks for .pfx,.p12 in trustpoints; .lic in license; and .rpm in rpms and aborts the current POAP if the checks/fileformats are not honoured.
- In case of .rpm, you need to provide the original file name in the yaml file.

For example: if you renamed customCliGoApp-1.0-1.7.5.x86_64.rpm to custom.rpm then PoAP will bail out indicating the name mismatch.

To get the original name of rpm:

```
bash-4.3$ rpm -qp --qf '%{NAME}-%{VERSION}-%{RELEASE}-%{ARCH}.rpm' custom.rpm
customCliGoApp-1.0-1.7.5.x86_64.rpm
bash-4.3$
```

- Once ISSU via POAP begins, abort of PoAP will be blocked. If ISSU fails for some reason, then abort capability will be re-enabled.

Guidelines and Limitations for POAP

POAP configuration guidelines and limitations are as follows:

- The bootflash:poap_retry_debugs.log is a file populated by POAP-PNP for internal purposes only. This file has no relevance in case of any POAP failures.
- The 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.

- The **https_ignore_certificate** option should be turned on to use the **ignore-certificate** keyword with https protocol in POAP. This would enable you to successfully perform HTTPS transfer in the POAP script and without this option https as protocol cannot work with POAP.
- If you use POAP to bootstrap a Cisco Nexus device that is a part of a virtual port channel (vPC) 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 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 9000 Series switch through a LACP port channel, the Cisco Nexus 9000 Series switch 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 9000 Series switch to not suspend its member ports by using the **no lacp suspend-individual** command from interface configuration mode.
- 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.
- You can bypass the password and the basic POAP configuration by using the **skip** option at the POAP prompt. When you use the **skip** option, no password is configured for the admin user. The **copy running-config startup-config** command is blocked until a valid password is set for the admin user.
- If the **boot poap enable** command (perpetual POAP) is enabled on the switch, on a reload, a POAP boot is triggered even if there is a startup configuration present. If you do not want to use POAP in this scenario, remove the boot poap enable configuration by using the **no boot poap enable** command.
- 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.
- Beginning with NX-OS 7.0(3)I7(4), RFC 3004 (User Class Option for DHCP) is supported. This enables POAP to support user-class option 77 for DHCPv4 and user-class option 15 for DHCPv6. The text displayed for the user class option for both DHCPv4 and DHCPv6 is "Cisco-POAP".
 - With RFC 3004 (User Class Option for DHCP) support, POAP over IPv6 is supported on Nexus 9000 switches.
 - Beginning with NX-OS 9.2(2), POAP over IPv6 is supported on Nexus 9504 and Nexus 9508 switches with -R line cards.

The POAP over IPv6 feature enables the POAP process to use IPv6 when IPv4 fails. The feature is designed to cycle between IPv4 and IPv6 protocols when a connection failure occurs.

- For secure POAP, ensure that DHCP snooping is enabled.
- To support POAP, set firewall rules to block unintended or malicious DHCP servers.
- To maintain system security and make POAP more secure, configure the following:
 - Enable DHCP snooping.
 - Set firewall rules to block unintended or malicious DHCP servers.
- POAP is supported on both MGMT ports and in-band ports.
- Beginning with Cisco NX-OS Release 10.6(2n)F, Cisco N9164E-NS4-O switch supports POAP over out-of-band (management) interface.

Setting Up the Network Environment to Use POAP

Procedure

-
- Step 1** Modify the basic configuration script provided by Cisco or create your own script. For information, see the *Python Scripting and API Configuration Guide*.
- Step 2** Every time you make a change to the configuration script, ensure that you recalculate the MD5 checksum by running `# f=poap_nexus_script.py ; cat $f | sed '/^#md5sum/d' > $f.md5 ; sed -i 's/^#md5sum=.*/#md5sum=\'$(md5sum $f.md5 | sed 's/ .*//')\''/' $f` using a bash shell. For more information, see the *Python API Reference Guide*.
- Step 3** (Optional) Put the POAP script and any other desired software image and switch configuration files on a USB device accessible to the switch.
- Step 4** 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 5** Deploy a TFTP or HTTP server to host the configuration script. In order to trigger the HTTP request to the server, prefix `HTTP://` to the TFTP server name. HTTPS is not supported.
- Step 6** Add the URL portion into the TFTP script name to show correct path to the file name.
- Step 7** 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.

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.

What to do next

Verify the configuration.

Verifying the Device Configuration

To verify the configuration, use one of the following commands:

Command	Purpose
show running-config	Displays the running configuration.
show startup-config	Displays the startup configuration.
	<p>Note</p> <p>If Layer 3 based feature configurations are disabled in the running-config, the show startup-config command does not display them. However, the configurations remain intact in the startup PSS, until the copy running startup command is performed.</p>

This option masks the sensitive words in running configuration output with <removed> keyword.



CHAPTER 5

Using Network Plug and Play

This chapter contains these sections:

- [About Network Plug and Play, on page 33](#)
- [Troubleshooting Examples for Network Plug and Play, on page 41](#)

About Network Plug and Play

Network Plug and Play (PnP) is a software application that runs on a Cisco Nexus 9500 Series Switch (specifically, N9K-C9504, N9K-C9508, and N9K-C9516). 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.
- 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 9500 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.

For information on configuring and managing the needed security certificate(s) for PnP functionality, see the [Cisco Digital Network Architecture Center Security Best Practices Guide](#).

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 is to provide consistent Day 0 deployment solution for all the deployment scenarios.

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

Day 0 Provisioning

Day 0 bootstrapping includes the configuration, image, and other files. 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 9500 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/DNA-C).

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.



Note POAP is the first order of choice for Day 0 provisioning. Only when there is no valid POAP offer, PnP discovery is attempted. Also, PnP is supported only on Cisco Nexus 9000 EoR models N9K-C9504, N9K-C9508, and N9K-C9516. PnP is not supported on Cisco Nexus 9000 ToRs.

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

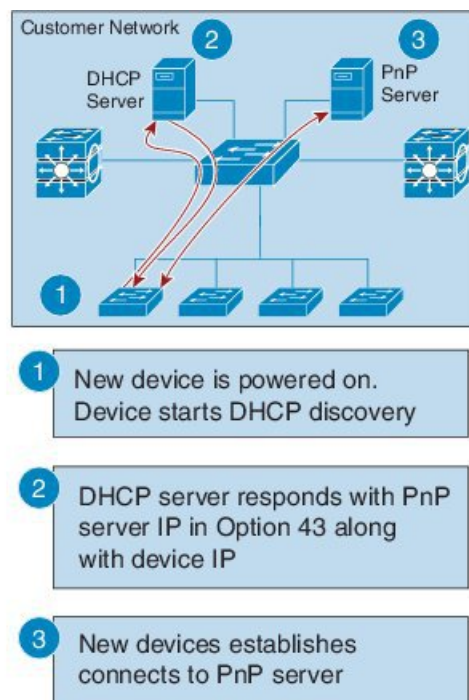
If the agent configuration fails, you should manually intervene and configure the switch.

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 server configuration.
- PnP agent handshakes with the PnP server.
- Download the image, install, and reload.
- Download and apply the configuration from the controller.

A device with no startup configuration in the NV-RAM triggers the day 0 provisioning and goes through the POAP process (as detailed in [m_using_poweron_auto_provisioning_92x.ditamap#id_70221](#)). When there is no valid POAP offer, the PnP agent is initiated. 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 6: DHCP Discovery Process for PnP Server



DNS-Based Discovery

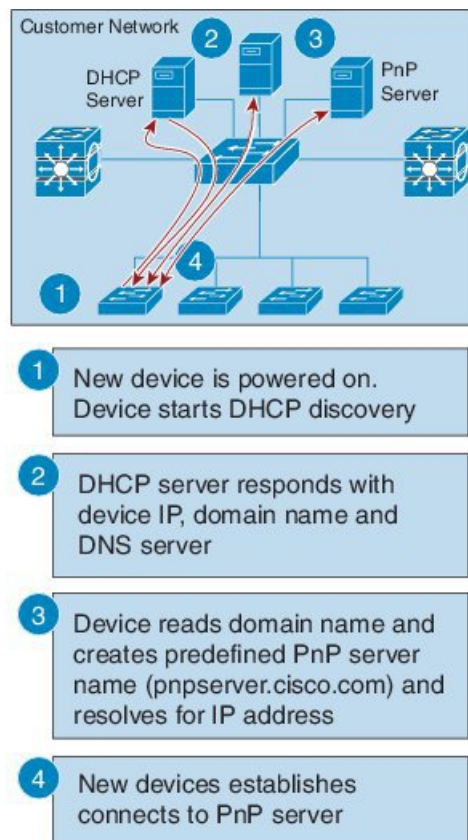
When the DHCP discovery fails to get the PnP server, the agent falls back to DNS-based discovery. To start the DNS-based discovery, the following information is required from DHCP:

- 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 7: DNS Lookup for `pnpserver.[domainname].com`



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 9500 Series Switch. PoAP and PnP interworking has the following processes:

- PoAP starts first when no start-up 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.

PnP Agent

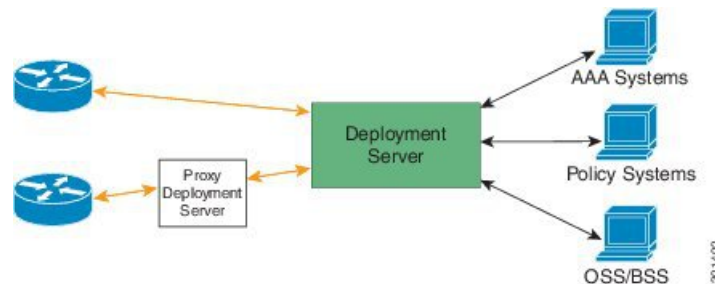
The PnP agent is an embedded software component that is present in all Cisco network devices that support simplified deployment architecture. The PnP agent understands and interacts only with a PnP server. The PnP agent first tries to discover a PnP server, with which it can communicate. After a server is found and connection established, the agent performs deployment-related activities such as configuration, image 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 8: 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.

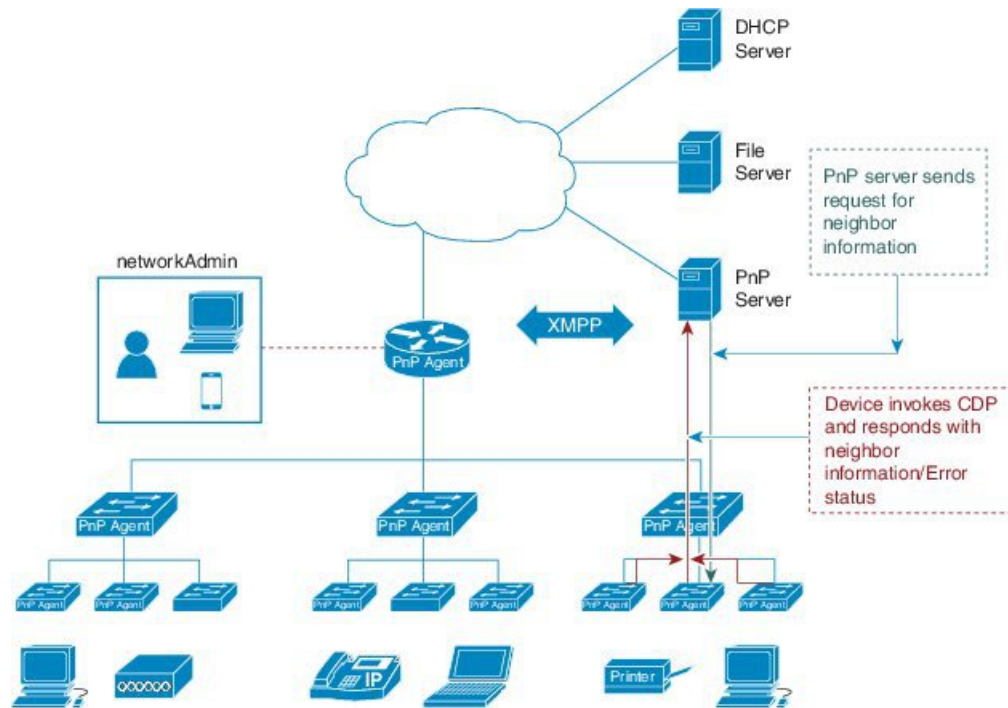
PnP Agent Deployment

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 9: Network Topology of Cisco PnP Agent Deployment



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.

Guidelines and Limitations for Network Plug and Play

Network Plug and Play (PnP) guidelines and limitations are as follows:

- Beginning with NX-OS 9.2(3), PnP is supported on the management port of Cisco Nexus 9500 platform switches.
- PnP runs on both the in-band and the management interfaces. In-band is supported only on FX-series line cards (specifically N9K-X9736C-FX for PnP).
- The PnP deployment method depends on the discovery process that is required for finding the PnP controller or server.
- The discovery mechanism must be deployed, either as a DHCP server discovery process or a Domain Name Server (DNS) discovery process, before launching PnP.

- Configure the DHCP server or the DNS server before deploying PnP.
- The PnP server must communicate with the PnP agent.
- PnP connect does not require a DHCP or DNS configuration.
- IPv6 support for PnP is not available for Cisco Nexus 9500 Series devices.

Cisco DNA Center Support

The following guidelines and limitations are specific for PnP connectivity to the Cisco DNA Center:

- Cisco DNA Center supports the following functionality on the Cisco Nexus 9504, Cisco Nexus 9508, and Cisco Nexus 9516 switches:
 - Discovery
 - Inventory
 - Topology
 - Template Programmer
 - Software Image Management
 - Basic Monitoring
 - The following PnP guidelines and limitations are only for the Cisco DNA Center version 1.2.6 and earlier:
 - The startup configuration that is provided during plug and play must ensure that the connectivity for the interface that is connected to the Cisco DNA Center remains intact.
 - The system image .bin and startup configuration must be uploaded to the Cisco DNA Center.
 - The bootflash must have enough space to download the image and configurations from the Cisco DNA Center.
- Click [here](#) for the user documentation for the Cisco DNA Center.

Troubleshooting Examples for Network Plug and Play

Example: Troubleshooting PnP

The following examples shows the PnP troubleshooting command outputs:

```
Switch# show pnp status
PnP Agent is running
server-connection
  status: Success
  time: 08:41:26 Jan 11
interface-info
  status: Success
  time: 08:34:00 Jan 11
device-info
  status: Success
  time: 08:33:46 Jan 11
config-upgrade
  status: Success
```

```

    time: 08:31:36 Jan 11
capability
  status: Success
    time: 08:33:50 Jan 11
backoff
  status: Success
    time: 08:41:26 Jan 11
topology
  status: Success
    time: 08:33:54 Jan 11

```

```

Switch# show pnp version
PnP Agent Version Summary

```

```

PnP Agent: 1.6.0
Platform Name: nxos
PnP Platform: 1.5.0.rc2

```

```

Switch# show pnp profiles
Created by          UDI
DHCP Discovery    PID:N9K-C9504,VID:V01,SN:FOX1813GCZ8

```

```

    Primary transport: https
    Address: 10.105.194.248
    Port: 443
    CA file: /etc/pnp/certs/trustpoint/pnplabel

```

```

    Work-Request Tracking:
      Pending-WR: Correlator=
Cisco-PnP-POSIX-nxos-1.6.0-21-589a466a-0d88-427b-a17e-69afb7d0a226-1
      Last-WR:    Correlator=
Cisco-PnP-POSIX-nxos-1.6.0-20-ab225de4-b0ef-46c5-9c4f-e3bd9f7c6b87-1
      PnP Response Tracking:
      Last-PR:    Correlator=
Cisco-PnP-POSIX-nxos-1.6.0-20-ab225de4-b0ef-46c5-9c4f-e3bd9f7c6b87-1

```

```

Switch# show pnp lease
{
  "lease": {
    "uptime": "Fri Jan 11 05:32:17 2019",
    "intf": "Vlan1",
    "ip_addr": "10.77.143.239",
    "mask": "255.255.255.0",
    "gw": "10.77.143.1",
    "domain": "",
    "opt_43": "5A1D;B2;K4;I10.105.194.248;J80",
    "lease": "3600",
    "server": "10.77.143.231",
    "vrf": "1"
  }
}

```

```

Switch# show pnp internal trace

```

- 1) Event:E_DEBUG, length:49, at 907122 usecs after Fri Jan 11 08:30:44 2019
[104] pnp_ascii_gen: ascii gen completed rcode[0]
- 2) Event:E_DEBUG, length:16, at 907094 usecs after Fri Jan 11 08:30:44 2019
[104] pss type: 5
- 3) Event:E_DEBUG, length:31, at 907069 usecs after Fri Jan 11 08:30:44 2019
[104] Entering pnp_ascii_gen_cfg

```
4) Event:E_DEBUG, length:22, at 907061 usecs after Fri Jan 11 08:30:44 2019
   [104] Calling Ascii gen

5) Event:E_DEBUG, length:16, at 907051 usecs after Fri Jan 11 08:30:44 2019
   [104] pss type: 2

6) Event:E_DEBUG, length:49, at 907018 usecs after Fri Jan 11 08:30:44 2019
   [104] pnp_ascii_gen: fu_num_acfg_pss_entries[0x2]

7) Event:E_DEBUG, length:49, at 973813 usecs after Fri Jan 11 08:29:51 2019
   [104] pnp_ascii_gen: ascii gen completed rcode[0]

8) Event:E_DEBUG, length:16, at 973787 usecs after Fri Jan 11 08:29:51 2019
   [104] pss type: 5

9) Event:E_DEBUG, length:31, at 973760 usecs after Fri Jan 11 08:29:51 2019
   [104] Entering pnp_ascii_gen_cfg

10) Event:E_DEBUG, length:22, at 973751 usecs after Fri Jan 11 08:29:51 2019
   [104] Calling Ascii gen

11) Event:E_DEBUG, length:16, at 973742 usecs after Fri Jan 11 08:29:51 2019
   [104] pss type: 2

12) Event:E_DEBUG, length:49, at 973707 usecs after Fri Jan 11 08:29:51 2019
   [104] pnp_ascii_gen: fu_num_acfg_pss_entries[0x2]

13) Event:E_DEBUG, length:35, at 535794 usecs after Fri Jan 11 08:04:15 2019
   [104] pnp_pi_spawn_finalize pid 690

14) Event:E_DEBUG, length:41, at 228291 usecs after Fri Jan 11 08:04:13 2019
   [104] + pnp_pi_spawn child_pid: 0xdd526da0

15) Event:E_DEBUG, length:76, at 132853 usecs after Fri Jan 11 08:03:26 2019
   [104] Rx: Direction: PnP PI -> PnP PD, Type: Device Provisioned, Cfg: Present

16) Event:E_DEBUG, length:35, at 440380 usecs after Fri Jan 11 08:03:18 2019
   [104] !!! ACKED Unconfigure Ret:1!!!

17) Event:E_DEBUG, length:61, at 440347 usecs after Fri Jan 11 08:03:18 2019
   [104] Tx: Direction: Max, Type: DHCP Unconfigure Done, Len: 16

18) Event:E_DEBUG, length:35, at 440331 usecs after Fri Jan 11 08:03:18 2019
   [102] Unknown timer cancel requested

19) Event:E_DEBUG, length:35, at 440311 usecs after Fri Jan 11 08:03:18 2019
   [104] pnp_pss_runtime_commit success

20) Event:E_DEBUG, length:57, at 440103 usecs after Fri Jan 11 08:03:18 2019
   [104] pnp_pss_runtime_commit: Stored values in runtime PSS

21) Event:E_DEBUG, length:23, at 440051 usecs after Fri Jan 11 08:03:18 2019
   [104] - pnp_vsh_halt:206

22) Event:E_DEBUG, length:17, at 950291 usecs after Fri Jan 11 08:03:15 2019
   [104] Adding "end"

23) Event:E_DEBUG, length:58, at 950269 usecs after Fri Jan 11 08:03:15 2019
   [104] Adding "configure terminal ; no clock protocol none "

24) Event:E_DEBUG, length:33, at 945994 usecs after Fri Jan 11 08:03:15 2019
   [104] - pnp_vsh_config_l3_intf:788
```

```
25) Event:E_DEBUG, length:29, at 945979 usecs after Fri Jan 11 08:03:15 2019
[104] + pnp_vsh_config_l3_intf

26) Event:E_DEBUG, length:39, at 945963 usecs after Fri Jan 11 08:03:15 2019
[104] Adding "no feature interface-vlan"

27) Event:E_DEBUG, length:32, at 945932 usecs after Fri Jan 11 08:03:15 2019
[104] Adding "configure terminal"

28) Event:E_DEBUG, length:40, at 945886 usecs after Fri Jan 11 08:03:15 2019
[104] Got Semaphore, vsh halt continue...

29) Event:E_DEBUG, length:46, at 945870 usecs after Fri Jan 11 08:03:15 2019
[104] sem_timedwait Success, Start VSH clean up

30) Event:E_DEBUG, length:19, at 945843 usecs after Fri Jan 11 08:03:15 2019
[104] + pnp_vsh_halt

31) Event:E_DEBUG, length:35, at 945831 usecs after Fri Jan 11 08:03:15 2019
[104] pnp_pss_runtime_commit success

32) Event:E_DEBUG, length:57, at 945643 usecs after Fri Jan 11 08:03:15 2019
[104] pnp_pss_runtime_commit: Stored values in runtime PSS

33) Event:E_DEBUG, length:33, at 945607 usecs after Fri Jan 11 08:03:15 2019
[104] !!! Received Unconfigure !!!

34) Event:E_DEBUG, length:74, at 945578 usecs after Fri Jan 11 08:03:15 2019
[104] Rx: Direction: PnP PI -> PnP PD, Type: DHCP Unconfigure, Cfg: Present

35) Event:E_DEBUG, length:49, at 789616 usecs after Fri Jan 11 08:01:52 2019
[104] pnp_ascii_gen: ascii gen completed rcode[0]

36) Event:E_DEBUG, length:16, at 789579 usecs after Fri Jan 11 08:01:52 2019
[104] pss type: 5

37) Event:E_DEBUG, length:31, at 789522 usecs after Fri Jan 11 08:01:52 2019
[104] Entering pnp_ascii_gen_cfg

38) Event:E_DEBUG, length:22, at 789514 usecs after Fri Jan 11 08:01:52 2019
[104] Calling Ascii gen

39) Event:E_DEBUG, length:16, at 789506 usecs after Fri Jan 11 08:01:52 2019
[104] pss type: 2

40) Event:E_DEBUG, length:49, at 789489 usecs after Fri Jan 11 08:01:52 2019
[104] pnp_ascii_gen: fu_num_acfg_pss_entries[0x2]

41) Event:E_DEBUG, length:35, at 789365 usecs after Fri Jan 11 08:01:52 2019
[104] pnp_pss_runtime_commit success

42) Event:E_DEBUG, length:57, at 789135 usecs after Fri Jan 11 08:01:52 2019
[104] pnp_pss_runtime_commit: Stored values in runtime PSS

43) Event:E_DEBUG, length:26, at 789096 usecs after Fri Jan 11 08:01:52 2019
[104] Phase Init -> Monitor

44) Event:E_DEBUG, length:35, at 788967 usecs after Fri Jan 11 08:01:52 2019
[104] pnp_pi_spawn_finalize pid 1c9

45) Event:E_DEBUG, length:41, at 831561 usecs after Fri Jan 11 08:01:49 2019
[104] + pnp_pi_spawn child_pid: 0xffff7e28

46) Event:E_DEBUG, length:45, at 831550 usecs after Fri Jan 11 08:01:49 2019
```

```
[104] Have startup config, Starting PnP PI....
47) Event:E_DEBUG, length:40, at 831538 usecs after Fri Jan 11 08:01:49 2019
[104] Posix log directory creation failed
48) Event:E_DEBUG, length:50, at 831479 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_fire_event: PNP_EVENT_HAVE_STARTUP_CONFIG
49) Event:E_DEBUG, length:35, at 831465 usecs after Fri Jan 11 08:01:49 2019
[104] Inside : pnp_other_msg_handler
50) Event:E_DEBUG, length:80, at 831437 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_get_data_from_queue: dequeued event 0x1102e0cc 25/cat 11 from pending Q
51) Event:E_DEBUG, length:50, at 831368 usecs after Fri Jan 11 08:01:49 2019
[104] Injecting Event PNP_EVENT_HAVE_STARTUP_CONFIG
52) Event:E_DEBUG, length:59, at 831303 usecs after Fri Jan 11 08:01:49 2019
[104] Have Startup Config, move the process state to monitor
53) Event:E_DEBUG, length:57, at 799379 usecs after Fri Jan 11 08:01:49 2019
[104] Accelerating PnP, Break Point: Break Point PoAP Init
54) Event:E_DEBUG, length:35, at 799334 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit success
55) Event:E_DEBUG, length:57, at 799239 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit: Stored values in runtime PSS
56) Event:E_DEBUG, length:23, at 799226 usecs after Fri Jan 11 08:01:49 2019
[104] Phase None -> Init
57) Event:E_DEBUG, length:53, at 799200 usecs after Fri Jan 11 08:01:49 2019
[104] Initalizing PnP-agent State machine curr_state 3
58) Event:E_DEBUG, length:35, at 799188 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit success
59) Event:E_DEBUG, length:57, at 799070 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit: Stored values in runtime PSS
60) Event:E_DEBUG, length:26, at 798965 usecs after Fri Jan 11 08:01:49 2019
[104] !!! Box is Online !!!
61) Event:E_DEBUG, length:35, at 798954 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit success
62) Event:E_DEBUG, length:57, at 798770 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit: Stored values in runtime PSS
63) Event:E_DEBUG, length:70, at 370297 usecs after Fri Jan 11 07:55:41 2019
[102] pnp_demux_mts(463): (Warning) unexpected mts msg (opcode - 7655)
64) Event:E_DEBUG, length:41, at 092701 usecs after Fri Jan 11 07:55:33 2019
[104] PnP Init Internal subsystem, Done!!!
65) Event:E_DEBUG, length:32, at 089920 usecs after Fri Jan 11 07:55:33 2019
[104] PnP Init Internal subsystem

Switch# show pnp posix_pi configs

/isan/etc/pnp/platform_config.cfg:

/isan/etc/pnp/file_paths.cfg:
```

```
/isan/etc/pnp/pnp_config.cfg:  
/isan/etc/pnp/policy_discovery.conf:  
/isan/etc/pnp/certs/platform.json:  
/isan/etc/pnp/certs/pnp_status.json:  
/isan/etc/pnp/certs/job_status.json:
```



CHAPTER 6

Understanding the Command-Line Interface

This chapter contains these sections:

- About the CLI Prompt, on page 47
- Command Modes, on page 48
- Special Characters, on page 52
- Keystroke Shortcuts, on page 53
- Abbreviating Commands, on page 56
- Completing a Partial Command Name, on page 56
- Identifying Your Location in the Command Hierarchy, on page 57
- Using the no Form of a Command, on page 57
- Configuring CLI Variables, on page 58
- Command Aliases, on page 60
- Command Scripts, on page 62
- Context-Sensitive Help, on page 64
- Understanding Regular Expressions, on page 66
- Searching and Filtering show Command Output, on page 67
- Searching and Filtering from the --More-- Prompt, on page 72
- Using the Command History, on page 73
- Enabling or Disabling the CLI Confirmation Prompts, on page 75
- Setting CLI Display Colors, on page 75
- Sending Commands to Modules, on page 76
- Sending Command Output in Email, on page 77
- BIOS Loader Prompt, on page 78
- Examples Using the CLI, on page 79

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 the following example:

```
User Access Verification
login: admin
Password:<password>
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
```

```
Copyright (c) 2002-2013, 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.

SUMMARY STEPS

1. **configure terminal**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	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 9000 Series NX-OS Interfaces Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	interface <i>type number</i> Example: <pre>switch(config)# interface ethernet 2/2 switch(config-if)#</pre>	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 9000 Series NX-OS Interfaces Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number.subint*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>type number.subint</i> 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 subinterface 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.

The following 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

SUMMARY STEPS

1. **exit**
2. **end**
3. (Optional) **Ctrl-Z**

DETAILED STEPS

Procedure		
	Command or Action	Purpose
Step 1	exit Example: <pre>switch(config-if) # exit switch(config) #</pre>	Exits from the current configuration command mode and returns to the previous configuration command mode.
Step 2	end Example: <pre>switch(config-if) # end switch#</pre>	Exits from the current configuration command mode and returns to EXEC mode.
Step 3	(Optional) Ctrl-Z Example: <pre>switch(config-if) # ^z switch#</pre>	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 4: 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 .

Mode	Access Method	Prompt	Exit Method
Interface configuration	From global configuration mode, specify an interface with an interface command.	<code>switch(config-if)#</code>	To exit to global configuration mode, use the exit command. To exit to EXEC mode, use the exit command or press Ctrl-Z .
Subinterface configuration	From global configuration mode, specify a subinterface with an interface command.	<code>switch(config-subif)#</code>	To exit to global configuration mode, use the exit command. To exit to EXEC mode, use the end command or press Ctrl-Z .
VRF configuration	From global configuration mode, use the vrf command and specify a routing protocol.	<code>switch(config-vrf)#</code>	To exit to global configuration mode, use the exit command. To exit to EXEC mode, use the end command or press Ctrl-Z .
EXEC for a nondefault VRF	From EXEC mode, use the routing-context vrf command and specify a VRF.	<code>switch-red#</code>	To exit to the default VRF, use the routing-context vrf default command.

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 5: 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 6: Keystroke Shortcuts

Keystrokes	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.
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 to the right one character.
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).

Keystrokes	Description
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.

Keystrokes	Description
Tab	<p>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.</p> <p>Use tabs to complete the following items:</p> <ul style="list-style-type: none"> • Command names • Scheme names in the file system • Server names in the file system • Filenames in the file system <p>Example:</p> <pre>switch(config)# xm<Tab> switch(config)# xml<Tab> switch(config)# xml server</pre> <p>Example:</p> <pre>switch(config)# c<Tab> callhome class-map clock cdp cli control-plane switch(config)# cl<Tab> class-map cli clock switch(config)# cla<Tab> switch(config)# class-map</pre> <p>Example:</p> <pre>switch# cd bootflash:<Tab> bootflash:/// bootflash://sup-1/ bootflash://sup-active/ bootflash://sup-local/ bootflash://module-27/ bootflash://module-28/</pre> <p>Example:</p> <pre>switch# cd bootflash://mo<Tab> bootflash://module-27/ bootflash://module-28/ switch# cd bootflash://module-2</pre> <p>Note You cannot access remote machines using the cd command. If you are on slot 27 and enter the cd bootflash://module-28 command, the following message appears: "Changing directory to a non-local server is not allowed."</p>

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 7: Examples of Command Abbreviations

Command	Abbreviation
configure terminal	conf t
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 the following 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 submenu hierarchy nested more than one level. In these cases, you can display information about your present working context (PWC).

SUMMARY STEPS

1. `where detail`

DETAILED STEPS

Procedure		
	Command or Action	Purpose
Step 1	<p><code>where detail</code></p> <p>Example:</p> <pre>switch# configure terminal switch(config)# interface mgmt0 switch(config-if)# where detail mode: conf interface mgmt0 username: admin routing-context vrf: default</pre>	Displays the PWC.

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.

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="2013-05-12-13.43.13"
testinterface="ethernet1/2"

switch# cli no var name testinterface
switch# show cli variables
SWITCHNAME="switch"
TIMESTAMP="2013-05-12-13.43.13"

```

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)`.

SUMMARY STEPS

1. **cli var name** *variable-name variable-text*
2. (Optional) **show cli variables**

DETAILED STEPS

Procedure

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

Configuring Persistent CLI Variables

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

SUMMARY STEPS

1. **configure terminal**
2. **cli var name** *variable-name variable-text*
3. **exit**
4. (Optional) **show cli variables**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	cli var name <i>variable-name variable-text</i> Example: <pre>switch(config)# cli var name testinterface ethernet 2/1</pre>	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: <pre>switch(config)# exit switch#</pre>	Exits global configuration mode.
Step 4	(Optional) show cli variables Example: <pre>switch# show cli variables</pre>	Displays the CLI variable configuration.
Step 5	(Optional) copy running-config startup-config Example: <pre>switch(config)# copy running-config startup-config</pre>	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.



Note When using the **cli alias name** command, few keywords are reserved and cannot be used. To view the list of reserved keywords, type the **show cli internal keywords common** command. The reserved keywords match either partially or fully. For instance, in the **cli alias name i show version** command, **i** is a partial match. To prevent the use of reserved keywords, enable strict checking using the **cli alias check strict** command. This command is not enabled by default. After enabling the command, if a user attempts to use any reserved keywords, the switch displays an error.

Defining Command Aliases

You can define command aliases for commonly used commands.

SUMMARY STEPS

1. **configure terminal**
2. **cli alias name alias-name alias-text**
3. **exit**
4. (Optional) **alias**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.

	Command or Action	Purpose
Step 2	cli alias name <i>alias-name alias-text</i> Example: switch(config)# cli alias name ethint interface ethernet	Configures the command alias. The alias name is an alphanumeric string that is not case 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.

SUMMARY STEPS

1. **terminal alias** [**persist**] *alias-name command-string*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	terminal alias [persist] <i>alias-name command-string</i> 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:` or `volatile:` directory on the Cisco NX-OS device.

SUMMARY STEPS

1. `run-script [bootflash: | volatile:] filename`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	<code>run-script [bootflash: volatile:] filename</code> Example: <pre>switch# run-script testfile</pre>	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 8: Formatting Options for the echo Command

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

Formatting Option	Description
<code>\nnn</code>	Displays the corresponding ASCII octal character.

SUMMARY STEPS

1. `echo [backslash-interpret] [text]`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	<code>echo [backslash-interpret] [text]</code> Example: <pre>switch# echo This is a test. This is a test.</pre>	The backslash-interpret keyword indicates that the text string contains formatting options. The <i>text</i> 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.

SUMMARY STEPS

1. `sleep seconds`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	<code>sleep seconds</code> Example: <pre>switch# sleep 30</pre>	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 9: Context-Sensitive Help Example

Example Outputs	Description
<pre>switch# clock ? set HH:MM:SS Current Time switch# clock</pre>	<p>Displays the command syntax for the clock command in EXEC mode.</p> <p>The switch output shows that the set keyword is required for using the clock command.</p>
<pre>switch# clock set ? WORD HH:MM:SS Current Time switch# clock set</pre>	<p>Displays the command syntax for setting the time.</p> <p>The help output shows that the current time is required for setting the clock and how to format the time.</p>
<pre>switch# clock set 13:32:00<CR> % Incomplete command switch#</pre>	<p>Adds the current time.</p> <p>The CLI indicates the command is incomplete.</p>
<pre>switch# <Ctrl-P> switch# clock set 13:32:00</pre>	<p>Displays the previous command that you entered.</p>
<pre>switch# clock set 13:32:00 ? <1-31> Day of the month switch# clock set 13:32:00</pre>	<p>Displays the additional arguments for the clock set command.</p>
<pre>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</pre>	<p>Displays the additional arguments for the clock set command.</p>
<pre>switch# clock set 13:32:00 18 April 13<CR> % Invalid input detected at '^' marker.</pre>	<p>Adds the date to the clock setting.</p> <p>The CLI indicates an error with the caret symbol (^) at 13.</p>
<pre>switch# clock set 13:32:00 18 April ? <2000-2030> Enter the year (no abbreviation) switch# clock set 13:32:00 18 April</pre>	<p>Displays the correct arguments for the year.</p>
<pre>switch# clock set 13:32:00 18 April 2013<CR> switch#</pre>	<p>Enters the correct syntax for the clock set command.</p>

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 10: 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.
^	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 11: Special Characters Used for Anchoring

Character	Description
<code>^</code>	Matches the beginning of the string.
<code>\$</code>	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
xml        Output in xml format (according to .xsd definitions)
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 12: Filtering and Searching Keywords

Keyword Syntax	Description
begin <i>string</i> Example: <code>show version begin Hardware</code>	Starts displaying at the line that contains the text that matches the search string. The search string is case sensitive.
count Example: <code>show running-config count</code>	Displays the number of lines in the command output.
cut [-d <i>character</i>] { -b -c -f -s } Example: <code>show file testoutput cut -b 1-10</code>	Displays only part of the output lines. You can display a number of bytes (-b), characters (-vcut [-d <i>character</i>] { -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 line that does not contain the delimiter.
end <i>string</i> Example: <code>show running-config end interface</code>	Displays all lines up to the last occurrence of the search string.
exclude <i>string</i> Example: <code>show interface brief exclude down</code>	Displays all lines that do not include the search string. The search string is case sensitive.
head [lines <i>lines</i>] Example: <code>show logging logfile head lines 50</code>	Displays the beginning of the output for the number of lines specified. The default number of lines is 10.

Keyword Syntax	Description
human Example: <pre>show version human</pre>	Displays the output in normal format if you have previously set the output format to XML using the terminal output xml command.
include <i>string</i> Example: <pre>show interface brief include up</pre>	Displays all lines that include the search string. The search string is case sensitive.
last [<i>lines</i>] Example: <pre>show logging logfile last 50</pre>	Displays the end of the output for the number of lines specified. The default number of lines is 10.
no-more Example: <pre>show interface brief no-more</pre>	Displays all the output without stopping at the end of the screen with the <code>--More--</code> prompt.
sscp <i>SSH-connection-name filename</i> Example: <pre>show version sscp MyConnection show_version_output</pre>	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 [<i>bytes lines words</i>] Example: <pre>show file testoutput wc bytes</pre>	Displays counts of characters, lines, or words. The default is to display the number of lines, words, and characters.
xml Example: <pre>show version xml</pre>	Displays the output in XML format.

diff Utility

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

diff-clean [**all-sessions**] [**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 13: 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.
<i>expression</i>	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.

Mini AWK Utility

AWK is a simple but powerful utility to summarize text output. You can use this utility after a pipe (|) to further process the text output of a command. Cisco NX-OS supports a mini AWK, which takes an inline program as an argument.

This example shows how the mini AWK utility can be used to summarize the text output of the **show ip route summary vrf all** command:

```
switch# show ip route summary vrf all | grep "Total number of routes"
Total number of routes: 3
Total number of routes: 10

switch# show ip route summary vrf all | grep "Total number of routes" | awk '{ x = x + $5}
END { print x }'
13
```

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 14: sort Utility Parameters

Parameter	Description
-M	Sorts by month.
-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 <i>field-number</i> [<i>.char-position</i>][<i>ordering</i>]	Sorts according to a key value. There is no default key value.

Parameter	Description
-n	Sorts according to a numeric string value.
-r	Reverses order of the sort results. The default sort output is in ascending order.
-t <i>delimiter</i>	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 15: --More-- Prompt Commands

Commands	Description
[<i>lines</i>]<space>	Displays output lines for either the specified number of lines or the current screen size.
[<i>lines</i>] z	Displays output lines for either the specified number of lines or the current screen size. If you use the <i>lines</i> argument, that value becomes the new default screen size.
[<i>lines</i>]<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 <i>lines</i> argument, that value becomes the new default number of lines to display for this command.
[<i>lines</i>] d or [<i>lines</i>]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 <i>lines</i> 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.
[<i>lines</i>] 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.
[<i>lines</i>] 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.

Commands	Description
<code>[count]/expression</code>	Skips to the line that matches the regular expression and displays a screen of output lines. Use the optional <i>count</i> 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.
<code>[count]n</code>	Skips to the next line that matches the current regular expression and displays a screen of output lines. Use the optional <i>count</i> argument to skip past matches.
<code>{! :!}[shell-cmd]}</code>	Executes the command specified in the <i>shell-cmd</i> argument in a subshell.
<code>.</code>	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`.

SUMMARY STEPS

1. `[no] terminal edit-mode vi [persist]`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	<code>[no] terminal edit-mode vi [persist]</code> Example: <pre>switch# terminal edit-mode vi</pre>	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:

show cli history [*lines*] [**config-mode** | **exec-mode** | **this-mode-only**] [**unformatted**]

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

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

```
switch# show cli history
```

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

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

This example shows how to display only the configuration commands in the command history:

```
switch(config)# show cli history config-mode
```

This example shows how to display only the EXEC commands in the command history:

```
switch(config)# show cli history exec-mode
```

This example shows how to display only the commands in the command history for the current command mode:

```
switch(config-if)# show cli history this-mode-only
```

This 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.

SUMMARY STEPS

1. `[no] terminal dont-ask [persist]`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	<code>[no] terminal dont-ask [persist]</code> Example: <pre>switch# terminal dont-ask</pre>	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 if the previous command failed.
- The user input displays in blue.
- The command output displays in the default color.

The default colors are sent by the terminal emulator software.

SUMMARY STEPS

1. `terminal color [evening] [persist]`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	<code>terminal color [evening] [persist]</code> Example:	Sets the CLI display colors for the terminal session. The evening keyword is not supported. The persist keyword

	Command or Action	Purpose
	switch# terminal color	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 27 show version | grep lc
```

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

```
switch# slot 27 quoted "show version" | diff
switch# slot 28 quoted "show version" | diff -c
*** /volatile/vsh_diff_1_root_8430_slot__quoted_show_version.old      Wed Apr 29 20:10:41
    2013
--- -   Wed Apr 29 20:10:41 2013
*****
*** 1,5 ****
! RAM 1036860 kB
! lc27
  Software
    BIOS:      version 6.20
    system:    version 6.1(2)I1(1) [build 6.1(2)]
--- 1,5 ----
! RAM 516692 kB
! lc28
  Software
    BIOS:      version 6.20
    system:    version 6.1(2)I1(1) [build 6.1(2)]
*****
*** 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)
```

Sending Command Output in Email

You can use the CLI to send the output of a **show** command to an email address using the pipe operator (|).



Note The email configuration remains persistent for all **show** command output until it is reconfigured.

When you upgrade from a release before Cisco NX-OS Release 9.3(3) to Cisco NX-OS Release 9.3(3) or later releases, email configuration will be missing. This is due to enabling DME functionality for this feature. To resolve this, you need to execute "no email" and reapply the entire email configuration.

SUMMARY STEPS

1. **configure terminal**
2. **email**
3. **smtp-host** *ip-address smtp-port port*
4. **vrf management**
5. **from** *email-address*
6. **reply-to** *email-address*
7. **exit**
8. **exit**
9. **show email**
10. *show-command* | **email subject** *subject email-address*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	email Example: <pre>switch(config)# email switch(config-email)#</pre>	Enters email configuration mode.
Step 3	smtp-host <i>ip-address smtp-port port</i> Example: <pre>switch(config-email)# smtp-host 198.51.100.1 smtp-port 25</pre>	Specifies the SMTP host IP address and the SMTP port number.

	Command or Action	Purpose
Step 4	vrf management Example: switch(config-email)# vrf management	Specifies a VRF for the email transmission.
Step 5	from email-address Example: switch(config-email)# from admin@Mycompany.com	Specifies the sender's email address.
Step 6	reply-to email-address Example: switch(config-email)# reply-to admin@Mycompany.com	Specifies the recipient's email address.
Step 7	exit Example: switch(config-email)# exit switch(config)#	Exits email configuration mode.
Step 8	exit Example: switch(config)# exit switch#	Exits global configuration mode.
Step 9	show email Example: switch# show email	Displays the email configuration.
Step 10	show-command email subject subject email-address Example: switch# show interface brief email subject show-interface admin@Mycompany.com Email sent	Uses the pipe operator () to send the output of the specified show command with a subject to an email address.

BIOS Loader Prompt

When the supervisor modules power up, a specialized BIOS image automatically loads and tries to locate a valid nx-os image for booting the system. If a valid nx-os 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 9000 Series NX-OS Troubleshooting Guide*.

Examples Using the CLI

This section includes examples of using the CLI.

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 2013  rcfg.2013-05-01-12.27.59

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

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

```

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)#

```

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

```

Sending Command Output in Email

This example shows how to send the output of the **show interface brief** command to an email address using the pipe operator (**|**):

```

switch(config)# email
switch(config-email)# smtp-host 198.51.100.1 smtp-port 25
switch(config-email)# vrf management
switch(config-email)# from admin@Mycompany.com
switch(config-email)# reply-to admin@Mycompany.com
switch(config-email)# exit
switch(config)# exit
switch# show email
SMTP host: 198.51.100.1
SMTP port: 25
Reply to: admin@Mycompany.com
From: admin@Mycompany.com
VRF: management
switch# show interface brief | email subject show-interface admin@Mycompany.com

Email sent

```

The email sent to `admin@Mycompany.com` with the subject "show-interface" shows the output of the command:

```

<snip>
-----
Ethernet  VLAN Type Mode   Status Reason           Speed   Port
Interface                                     Ch #
-----
Eth1/1    --  eth  trunk  up      none              10G (D)  --
Eth1/2    --  eth  routed down  Link not connected auto (D)  --
Eth1/3    --  eth  routed up   none              10G (D)  --
Eth1/4    --  eth  routed down Link not connected auto (D)  --
Eth1/5    --  eth  routed down Link not connected auto (D)  --
Eth1/6    --  eth  routed down Link not connected auto (D)  --
Eth1/7    --  eth  routed down Link not connected auto (D)  --
Eth1/8    --  eth  routed down Link not connected auto (D)  --
Eth1/9    --  eth  routed down Link not connected auto (D)  --
Eth1/10   --  eth  routed down Link not connected auto (D)  --
<snip>

```




CHAPTER 7

Configuring Terminal Settings and Sessions

This chapter contains these sections:

- [About Terminal Settings and Sessions, on page 83](#)
- [Default Settings for File System Parameters, on page 84](#)
- [Configuring the Console Port, on page 84](#)
- [Configuring Virtual Terminals, on page 86](#)
- [Clearing Terminal Sessions, on page 88](#)
- [Displaying Terminal and Session Information, on page 88](#)

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.

Virtual Terminals

You can use virtual terminal lines to connect to your 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.

Default Settings for File System Parameters

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

Table 16: Default File System Settings

Parameters	Default
Default filesystem	bootflash:

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.

SUMMARY STEPS

1. **configure terminal**
2. **line console**
3. **databits** *bits*
4. **exec-timeout** *minutes*
5. **parity** {*even* | *none* | *odd*}

6. **speed** {300 | 1200 | 2400 | 4800 | 9600 | 38400 | 57600 | 115200}
7. **stopbits** {1 | 2}
8. **exit**
9. (Optional) **show line console**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	line console Example: <pre>switch# line console switch(config-console)#</pre>	Enters console configuration mode.
Step 3	databits <i>bits</i> Example: <pre>switch(config-console)# databits 7</pre>	Configures the number of data bits per byte. The range is from 5 to 8. The default is 8.
Step 4	exec-timeout <i>minutes</i> Example: <pre>switch(config-console)# exec-timeout 30</pre>	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: <pre>switch(config-console)# parity even</pre>	Configures the parity. The default is none .
Step 6	speed {300 1200 2400 4800 9600 38400 57600 115200} Example: <pre>switch(config-console)# speed 115200</pre>	Configures the transmit and receive speed. The default is 9600.
Step 7	stopbits {1 2} Example: <pre>switch(config-console)# stopbits 2</pre>	Configures the stop bits. The default is 1.
Step 8	exit Example: <pre>switch(config-console)# exit switch(config)#</pre>	Exits console configuration mode.

	Command or Action	Purpose
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: 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 the device.

SUMMARY STEPS

1. **configure terminal**
2. **line vty**
3. **exec-timeout *minutes***
4. **exit**
5. (Optional) **show running-config all | begin vty**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

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.
Step 3	exec-timeout <i>minutes</i> Example: switch(config-line)# exec-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.

	Command or Action	Purpose
Step 4	exit Example: <pre>switch(config-line)# exit switch(config)#</pre>	Exits line configuration mode.
Step 5	(Optional) show running-config all begin vty Example: <pre>switch(config)# show running-config all begin vty</pre>	Displays the virtual terminal configuration.
Step 6	(Optional) copy running-config startup-config Example: <pre>switch(config)# copy running-config startup-config</pre>	Copies the running configuration to the startup configuration.

Configuring the Session Limit

You can limit the number of virtual terminal sessions on your device.

SUMMARY STEPS

1. **configure terminal**
2. **line vty**
3. **session-limit sessions**
4. **exit**
5. (Optional) **show running-config all | begin vty**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	line vty Example: <pre>switch# line vty switch(config-line)#</pre>	Enters line configuration mode.
Step 3	session-limit sessions Example:	Configures the maximum number of virtual sessions for your device. The range is from 1 to 64. The default is 32.

	Command or Action	Purpose
	<code>switch(config-line)# session-limit 10</code>	
Step 4	exit Example: <code>switch(config-line)# exit</code> <code>switch(config)#</code>	Exits line configuration mode.
Step 5	(Optional) show running-config all begin vty Example: <code>switch(config)# show running-config all begin vty</code>	Displays the virtual terminal configuration.
Step 6	(Optional) copy running-config startup-config Example: <code>switch(config)# copy running-config startup-config</code>	Copies the running configuration to the startup configuration.

Clearing Terminal Sessions

You can clear terminal sessions on your device.

SUMMARY STEPS

1. (Optional) **show users**
2. **clear line** *name*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) show users Example: <code>switch# show users</code>	Displays the user sessions on the device.
Step 2	clear line <i>name</i> Example: <code>switch# clear line pts/0</code>	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.



CHAPTER 8

Basic Device Management

This chapter contains these sections:

- [About Basic Device Management, on page 91](#)
- [Default Settings for Basic Device Parameters, on page 92](#)
- [Changing the Device Hostname, on page 92](#)
- [Configuring the MOTD Banner, on page 93](#)
- [Configuring the Time Zone, on page 95](#)
- [Configuring Summer Time \(Daylight Saving Time\), on page 96](#)
- [Manually Setting the Device Clock, on page 97](#)
- [Setting the Clock Manager, on page 97](#)
- [Managing Users, on page 98](#)
- [Verifying the Device Configuration, on page 99](#)

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.

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 NX-OS device might contain clocks of different types that might need to be synchronized. These clocks are a part of various components (such as the supervisor, line card processors, or line cards), and each might 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 9000 Series NX-OS Security Configuration Guide*.

Default Settings for Basic Device Parameters

This table lists the default settings for basic device parameters.

Table 17: Default Basic Device Parameters

Parameters	Default
MOTD banner text	User Access Verification
Clock time zone	UTC

Changing the Device Hostname

You can change the device hostname displayed in the command prompt from the default (switch) to another character string.

SUMMARY STEPS

1. **configure terminal**
2. **{ hostname | switchname } name**
3. **exit**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	{hostname switchname} name Example: Using the hostname command: <pre>switch(config)# hostname Engineering1 Engineering1(config)#</pre> Using the switchname command: <pre>Engineering1(config)# switchname Engineering2 Engineering2(config)#</pre>	Changes the device hostname. The <i>name</i> argument is alphanumeric and case sensitive. The default is switch. Note The switchname command performs the same function as the hostname command. Beginning with Cisco NX-OS Release 7.0(3)I7(3), a maximum length of 63 characters for the switchname is supported.
Step 3	exit Example: <pre>Engineering2(config)# exit Engineering2#</pre>	Exits global configuration mode.
Step 4	(Optional) copy running-config startup-config Example: <pre>Engineering2# copy running-config startup-config</pre>	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 80 characters per line
- Maximum of 40 lines

SUMMARY STEPS

1. **configure terminal**
2. **banner motd** *delimiting-character message delimiting-character*
3. **exit**
4. (Optional) **show banner motd**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	banner motd <i>delimiting-character message delimiting-character</i> Example: <pre>switch(config)# banner motd #Welcome to the Switch# switch(config)#</pre>	Configures the MOTD banner. Do not use the <i>delimiting-character</i> in the <i>message</i> text. Note Do not use " or % as a delimiting character. Ensure that when you upgrade to higher releases for the limitation of this feature. Note Beginning from Cisco NX-OS Release 10.1(x), the following special characters (" , % , > , < , ' , (space), and ASCII characters < 0x15) are invalid as delimiting characters. If an existing MOTD banner with these delimiting characters is edited or a fresh banner is added with these delimiting characters, the banner is not configured to the running configuration. When you upgrade from an earlier release i.e, before 10.x releases to an existing 10.x releases, there is no impact on the configuration in the CLI and the configuration will be the same in the running configuration.
Step 3	exit Example: <pre>switch(config)# exit switch#</pre>	Exits global configuration mode.
Step 4	(Optional) show banner motd Example: <pre>switch# show banner motd</pre>	Displays the configured MOTD banner.
Step 5	(Optional) copy running-config startup-config Example: <pre>switch# copy running-config startup-config</pre>	Copies the running configuration to the startup configuration.

Configuring the Time Zone

You can configure the time zone to offset the device clock time from UTC.

SUMMARY STEPS

1. **configure terminal**
2. **clock timezone** *zone-name offset-hours offset-minutes*
3. **exit**
4. (Optional) **show clock**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	clock timezone <i>zone-name offset-hours offset-minutes</i> Example: <pre>switch(config)# clock timezone EST -5 0</pre>	Configures the time zone. The <i>zone-name</i> argument is a 3(or more)-character string(Max Size 8) for the time zone acronym (for example, PST or EST). The <i>offset-hours</i> argument is the offset from the UTC and the range is from -14 to 12 hours. The range for the <i>offset-minutes</i> argument is from 0 to 59 minutes.
Step 3	exit Example: <pre>switch(config)# exit switch#</pre>	Exits global configuration mode.
Step 4	(Optional) show clock Example: <pre>switch# show clock</pre>	Displays the time and time zone.
Step 5	(Optional) copy running-config startup-config Example: <pre>switch# copy running-config startup-config</pre>	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.

SUMMARY STEPS

1. **configure terminal**
2. **clock summer-time** *zone-name start-week start-day start-month start-time end-week end-day end-month end-time offset-minutes*
3. **exit**
4. (Optional) **show clock detail**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	clock summer-time <i>zone-name start-week start-day start-month start-time end-week end-day end-month end-time offset-minutes</i> Example: <pre>switch(config)# clock summer-time PDT 1 Sunday March 02:00 1 Sunday November 02:00 60</pre>	<p>Configures summer time or daylight saving time.</p> <p>The <i>zone-name</i> argument is a three character string for the time zone acronym (for example, PST and EST).</p> <p>The values for the <i>start-day</i> and <i>end-day</i> arguments are Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday.</p> <p>The values for the <i>start-month</i> and <i>end-month</i> arguments are January, February, March, April, May, June, July, August, September, October, November, and December.</p> <p>The value for the <i>start-time</i> and <i>end-time</i> arguments are in the format <i>hh:mm</i>.</p> <p>The range for the <i>offset-minutes</i> argument is from 0 to 1440 minutes.</p>
Step 3	exit Example: <pre>switch(config)# exit switch#</pre>	Exits global configuration mode.
Step 4	(Optional) show clock detail Example:	Displays the configured MOTD banner.

	Command or Action	Purpose
	<code>switch(config)# show clock detail</code>	
Step 5	(Optional) copy running-config startup-config Example: <code>switch# copy running-config startup-config</code>	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.

SUMMARY STEPS

1. **clock set** *time day month year*
2. (Optional) **show clock**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	clock set <i>time day month year</i> Example: <code>switch# clock set 15:00:00 30 May 2013</code> <code>Fri May 30 15:14:00 PDT 2013</code>	Configures the device clock. The format for the <i>time</i> argument is <i>hh:mm:ss</i> . The range for the <i>day</i> argument is from 1 to 31. The values for the <i>month</i> argument are January, February, March, April, May, June, July, August, September, October, November, and December . The range for the <i>year</i> argument is from 2000 to 2030.
Step 2	(Optional) show clock Example: <code>switch(config)# show clock</code>	Displays the current clock value.

Setting the Clock Manager

You can configure the clock manager to synchronize all the clocks of the components in the Cisco Nexus device.

SUMMARY STEPS

1. `clock protocol protocol`
2. (Optional) `show run clock_manager`

DETAILED STEPS**Procedure**

	Command or Action	Purpose
Step 1	<p><code>clock protocol protocol</code></p> <p>Example:</p> <pre>switch# clock protocol ntp</pre>	<p>Configures the clock manager.</p> <p>The values for the <i>protocol</i> argument are ntp and none.</p> <p>The following describes the values:</p> <ul style="list-style-type: none"> • ntp—Synchronizes clocks with Network Time Protocol (NTP). • none—Uses <code>clock set HH:MM:SS</code> to set the supervisor clock. <p>Note When none is used, the clock must be configured.</p> <p>Note Once the protocol is configured, the clock must use that protocol.</p>
Step 2	<p>(Optional) <code>show run clock_manager</code></p> <p>Example:</p> <pre>switch# show run clock_manager</pre>	<p>Displays the configuration of the clock manager.</p>

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.

SUMMARY STEPS

1. `show users`

DETAILED STEPS

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.

SUMMARY STEPS

1. (Optional) **show users**
2. **send [session line] message-text**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) show users Example: switch# show users	Displays the active user sessions.
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, use one of the following commands:

Command	Purpose
show running-config	Displays the running configuration.
show startup-config	Displays the startup configuration. Note If Layer 3 based feature configurations are disabled in the running-config, the show startup-config command does not display them. However, the configurations remain intact in the startup PSS, until the copy running startup command is performed.

This option masks the sensitive words in running configuration output with <removed> keyword.



CHAPTER 9

Using the Device File Systems, Directories, and Files

This chapter contains these sections:

- [About the Device File Systems, Directories, and Files, on page 101](#)
- [Default Settings for File System Parameters, on page 103](#)
- [Configuring the FTP, HTTP, or TFTP Source Interface, on page 103](#)
- [Working with Directories, on page 104](#)
- [Working with Files, on page 106](#)
- [Working with Archive Files, on page 113](#)
- [Examples of Using the File System, on page 116](#)

About the Device File Systems, Directories, and Files

This section describes file systems, directories, and files on the Cisco NX-OS device.

File Systems

The syntax for specifying a local file system is `filesystem:[/modules/]`.

This table describes file systems that you can reference on your device.

Table 18: File System Syntax Components

File System Name	Module	Description
bootflash	sup-active sup-local	Internal CompactFlash memory located on the 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 the 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 the active supervisor that stores logging file statistics.
system	—	Memory on a supervisor module used for storing the running-configuration file.
debug	—	Memory on a supervisor module used for debug logs.

Directories

You can create directories on bootflash: and external flash memory (usb1: and usb2:). You can navigate through these directories and use them for files.

Files

You create and access files on bootflash:, volatile:, usb1:, and usb2: filesystems. You can only access files on the system: filesystem. You can use the log: filesystem for debug log files.

You can download files, such as the nx-os image file, from remote servers using FTP, Secure Copy (SCP), Secure Shell FTP (SFTP), and TFTP. You can also copy files from an external server to the device, because the device can act as an SCP server.

Default Settings for File System Parameters

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

Table 19: Default File System Settings

Parameters	Default
Default filesystem	bootflash:

Configuring the FTP, HTTP, or TFTP Source Interface

You can configure the source interface for the File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), or Trivial File Transfer Protocol (TFTP). This configuration allows you to use the IP address associated with the configured source interface when copy packets are transferred.

SUMMARY STEPS

1. **configure terminal**
2. **[no] ip {ftp | http | tftp} source-interface {ethernet slot/port | loopback number}**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure		
	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	[no] ip {ftp http tftp} source-interface {ethernet slot/port loopback number} Example: <pre>switch(config)# ip tftp source-interface ethernet 2/1</pre>	Configures the source interface for all FTP, HTTP, or TFTP packets.
Step 3	(Optional) copy running-config startup-config Example: <pre>switch(config)# copy running-config startup-config</pre>	Copies the running configuration to the startup configuration.

Working with Directories

This section describes how to work with directories on the Cisco NX-OS device.

Identifying the Current Directory

You can display the directory name of your current directory.

SUMMARY STEPS

1. `pwd`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	<p><code>pwd</code></p> <p>Example:</p> <pre>switch# pwd</pre>	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:`.

SUMMARY STEPS

1. (Optional) `pwd`
2. `cd {directory | filesystem:[//module/][directory]}`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	<p>(Optional) <code>pwd</code></p> <p>Example:</p> <pre>switch# pwd</pre>	Displays the name of your current default directory.
Step 2	<p><code>cd {directory filesystem:[//module/][directory]}</code></p> <p>Example:</p> <pre>switch# cd usb1:</pre>	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.

SUMMARY STEPS

1. (Optional) **pwd**
2. (Optional) **cd** *{directory | filesystem:[//module/][directory]}*
3. **mkdir** *[filesystem:[//module/]]directory*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) pwd Example: switch# pwd	Displays the name of your current default directory.
Step 2	(Optional) cd <i>{directory filesystem:[//module/][directory]}</i> Example: switch# cd slot0:	Changes to a new current directory. The file system, module, and directory names are case sensitive.
Step 3	mkdir <i>[filesystem:[//module/]]directory</i> Example: switch# mkdir test	Creates a new directory. The <i>filesystem</i> argument is case sensitive. The <i>directory</i> argument is alphanumeric, case sensitive, and has a maximum of 64 characters.

Displaying Directory Contents

You can display the contents of a directory.

SUMMARY STEPS

1. **dir** *[directory | filesystem:[//module/][directory]]*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	dir <i>[directory filesystem:[//module/][directory]]</i> 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.

SUMMARY STEPS

1. (Optional) **pwd**
2. (Optional) **dir** *[filesystem :[/module/][directory]]*
3. **rmdir** *[filesystem :[/module/]]directory*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) pwd Example: switch# pwd	Displays the name of your current default directory.
Step 2	(Optional) dir <i>[filesystem :[/module/][directory]]</i> 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 <i>[filesystem :[/module/]]directory</i> Example: switch# rmdir test	Deletes a directory. The file system and directory name are case sensitive.

Accessing Directories on the Standby Supervisor Module

You can access all file systems on the standby supervisor module (remote) from a session on the active supervisor module. This feature is useful when copying files to the active supervisor modules requires similar files to exist on the standby supervisor module. To access the file systems on the standby supervisor module from a session on the active supervisor module, you specify the standby supervisor module in the path to the file using either *filesystem://sup-remote/* or *filesystem://sup-standby/*.

Working with Files

This section describes how to work with files on the 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.

SUMMARY STEPS

1. (Optional) **pwd**
2. (Optional) **dir** *[filesystem://module][directory]*
3. **move** *[filesystem://module][directory /] | directory/**source-filename* { *{filesystem://module}[directory /] | directory/* } *[target-filename] | target-filename* }

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) pwd Example: switch# pwd	Displays the name of your current default directory.
Step 2	(Optional) dir <i>[filesystem://module][directory]</i> Example: switch# dir bootflash	Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 3	move <i>[filesystem://module][directory /] directory/</i> <i>source-filename</i> { <i>{filesystem://module}[directory /] directory/</i> } <i>[target-filename] target-filename</i> } Example: switch# move test old_tests/test1	Moves a file. The file system, module, and directory names are case sensitive. The <i>target-filename</i> argument is alphanumeric, case sensitive, and has a maximum of 64 characters. If the <i>target-filename</i> argument is not specified, the filename defaults to the <i>source-filename</i> argument value.

Copying Files

You can make copies of files, either within the same directory or on another directory. For more information, see the *Cisco Nexus 9000 Series NX-OS Troubleshooting Guide*.



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.

SUMMARY STEPS

1. (Optional) **pwd**
2. (Optional) **dir** [*filesystem://module/*][*directory*]
3. **copy** [*filesystem://module/*][*directory/*] | *directory/*]*source-filename* | [*filesystem://module/*][*directory/*] | *directory/*}[*target-filename*]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) pwd Example: switch# pwd	Displays the name of your current default directory.
Step 2	(Optional) dir [<i>filesystem://module/</i>][<i>directory</i>] Example: switch# dir bootflash	Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 3	copy [<i>filesystem://module/</i>][<i>directory/</i>] <i>directory/</i>] <i>source-filename</i> [<i>filesystem://module/</i>][<i>directory/</i>] <i>directory/</i> }[<i>target-filename</i>] Example: switch# copy test old_tests/test1	Copies a file. The file system, module, and directory names are case sensitive. The <i>source-filename</i> argument is alphanumeric, case sensitive, and has a maximum of 64 characters. If the <i>target-filename</i> argument is not specified, the filename defaults to the <i>source-filename</i> argument value.

Deleting Files

You can delete a file from a directory.

SUMMARY STEPS

1. (Optional) **dir** [*filesystem://module/*][*directory*]
2. **delete** [*filesystem://module/*][*directory/*] | *directory/*]*filename*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) dir [<i>filesystem</i> :[// <i>module</i>]/][<i>directory</i>] Example: switch# dir bootflash:	Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 2	delete { <i>filesystem</i> :[// <i>module</i>]/}[<i>directory</i> /] <i>filename</i> Example: switch# delete bootflash:old_config.cfg	Deletes a file. The file system, module, and directory names are case sensitive. The <i>source-filename</i> 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.

SUMMARY STEPS

1. **show file** [*filesystem*:[//*module*]/][*directory*/]*filename*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	show file [<i>filesystem</i> :[// <i>module</i>]/][<i>directory</i> /] <i>filename</i> Example: switch# show file bootflash:test-results	Displays the file contents.

Displaying File Checksums

You can display checksums to check the file integrity.

SUMMARY STEPS

1. **show file** [*filesystem*:[//*module*]/][*directory*/]*filename* {**cksum** | **md5sum**}

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	show file [<i>filesystem:[//module/]</i>][<i>directory/</i>] <i>filename</i> { <i>cksum</i> <i>md5sum</i> } 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 device using Lempel-Ziv 1977 (LZ77) coding.

SUMMARY STEPS

1. (Optional) **dir** [*filesystem:[//module/]*]*directory*]
2. **gzip** [*filesystem:[//module/]*][*directory/*] | *directory/*]*filename*
3. **gunzip** [*filesystem:[//module/]*][*directory/*] | *directory/*]*filename* .gz

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) dir [<i>filesystem:[//module/]</i>] <i>directory</i>] Example: switch# dir bootflash:	Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 2	gzip [<i>filesystem:[//module/]</i>][<i>directory/</i>] <i>directory/</i>] <i>filename</i> Example: switch# gzip show_tech	Compresses a file. After the file is compressed, it has a .gz suffix.
Step 3	gunzip [<i>filesystem:[//module/]</i>][<i>directory/</i>] <i>directory/</i>] <i>filename</i> .gz Example: switch# gunzip show_tech.gz	Uncompresses a file. The file to uncompress must have the .gz suffix. After the file is uncompresses, it does not have the .gz suffix.

Displaying the Last Lines in a File

You can display the last lines of a file.

SUMMARY STEPS

1. **tail** [*filesystem:[//module/]*][*directory/*]*filename* [*lines*]

DETAILED STEPS**Procedure**

	Command or Action	Purpose
Step 1	tail [<i>filesystem:[//module/]</i>][<i>directory/</i>] <i>filename</i> [<i>lines</i>] 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:, volatile:, or a remote server. You can also specify the format for the command output.

SUMMARY STEPS

1. (Optional) **terminal redirection-mode** {**ascii** | **zipped**}
2. *show-command* > [*filesystem:[//module/]*][*directory*] | [*directory /*]*filename*

DETAILED STEPS**Procedure**

	Command or Action	Purpose
Step 1	(Optional) terminal redirection-mode { ascii zipped } Example: switch# terminal redirection-mode zipped	Sets the redirection mode for the show command output for the user session. The default mode is ascii .
Step 2	<i>show-command</i> > [<i>filesystem:[//module/]</i>][<i>directory</i>] [<i>directory /</i>] <i>filename</i> Example: switch# show tech-support > bootflash:techinfo	Redirects the output from a show command to a file.

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.

SUMMARY STEPS

1. (Optional) **pwd**

2. (Optional) `cd {filesystem:[//module/][directory] | directory}`
3. `find filename-prefix`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) <code>pwd</code> Example: switch# pwd	Displays the name of your current default directory.
Step 2	(Optional) <code>cd {filesystem:[//module/][directory] directory}</code> Example: switch# cd bootflash:test_scripts	Changes the default directory.
Step 3	<code>find filename-prefix</code> 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.

Formatting the Bootflash

Use the **format bootflash:** CLI command to format the onboard flash memory (bootflash:). If the command errors out due to the `Deactivate all virtual-services` and `try again` error message, destroy the Guest Shell using the **guestshell destroy** CLI command and rerun the **format bootflash:** command, for example,

```
switch# sh virtual-service list
Virtual Service List:
```

```

Name                               Status           Package Name
-----
guestshell+                         Activated       guestshell.ova
```

```
switch#
```

```
switch# guestshell destroy
You are about to destroy the guest shell and all of its contents. Be sure to save your
work. Are you sure you want to continue? (y/n) [n] y
```

```
switch# 2018 Jan 17 18:42:24 switch %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Deactivating
virtual service 'guestshell+'
```

```
switch#format bootflash:
```

Working with Archive Files

The Cisco NX-OS software supports archive files. You can create an archive file, append files to an existing archive file, extract files from an archive file, and list the files in an archive file.

Creating an Archive File

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.

SUMMARY STEPS

1. **tar create** {bootflash: | volatile:}archive-filename [absolute] [bz2-compress] [gz-compress] [remove] [uncompressed] [verbose] filename-list

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	<p>tar create {bootflash: volatile:}archive-filename [absolute] [bz2-compress] [gz-compress] [remove] [uncompressed] [verbose] filename-list</p> <p>Example:</p> <pre>switch# tar create bootflash:config-archive gz-compress bootflash:config-file</pre>	<p>Creates an archive file and adds files to it. The filename is alphanumeric, not case sensitive, and has a maximum length of 240 characters.</p> <p>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.</p> <p>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:</p> <ul style="list-style-type: none"> • For bz2-compress, the extension is .tar.bz2. • For gz-compress, the extension is .tar.gz. • For uncompressed, the extension is .tar. <p>The remove keyword specifies that the Cisco NX-OS software should delete the files from the file system after</p>

	Command or Action	Purpose
		<p>adding them to the archive. By default, the files are not deleted.</p> <p>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.</p>

Appending Files to an Archive File

You can append files to an existing archive file on your device.

Before you begin

You have created an archive file on your device.

SUMMARY STEPS

1. `tar append {bootflash: | volatile;}archive-filename [absolute] [remove] [verbose] filename-list`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	<code>tar append {bootflash: volatile;}archive-filename [absolute] [remove] [verbose] filename-list</code>	<p>Adds files to an existing archive file. The archive filename is not case sensitive.</p> <p>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.</p> <p>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.</p> <p>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.</p>

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 device.

Before you begin

You have created an archive file on your device.

SUMMARY STEPS

1. `tar extract {bootflash: | volatile:}archive-filename [keep-old] [screen] [to {bootflash: | volatile:}[/directory-name]] [verbose]`

DETAILED STEPS

Procedure		
	Command or Action	Purpose
Step 1	<pre>tar extract {bootflash: volatile:}archive-filename [keep-old] [screen] [to {bootflash: volatile:}[/directory-name]] [verbose]</pre> <p>Example:</p> <pre>switch# tar extract bootflash:config-archive.tar.gz</pre>	<p>Extracts files from an existing archive file. The archive filename is not case sensitive.</p> <p>The keep-old keyword indicates that the Cisco NX-OS software should not overwrite files with the same name as the files being extracted.</p> <p>The screen keyword indicates that the Cisco NX-OS software should not overwrite files with the same name as the files being extracted.</p> <p>The to keyword specifies the target filesystem. You can include a directory name. The directory name is alphanumeric, case sensitive, and has a maximum length of 240 characters.</p> <p>The verbose keyword specifies that the Cisco NX-OS software should display the names of the files as they are extracted.</p>

Displaying the Filenames in an Archive File

You can display the names of the files in an archive files using the **tar list** command.

```
tar list {bootflash: | volatile:}archive-filename
```

The archive filename is not case sensitive.

```
switch# tar list bootflash:config-archive.tar.gz
config-file
new-config
```

Examples of Using the File System

This section includes examples of how to use the file system on the Cisco NX-OS device.

Accessing Directories on Standby Supervisor Modules

This example shows how to list the files on the standby supervisor module:

```
switch# dir bootflash://sup-remote
 4096   Oct 03 23:55:55 2013  .patch/
...
 16384   Jan 01 13:23:30 2011  lost+found/
 297054208 Oct 21 18:55:36 2013  n9000-dk9.6.1.2.I1.1.bin
...

Usage for bootflash://sup-remote
1903616000 bytes used
19234234368 bytes free
21137850368 bytes total
```

This example shows how to delete a file on the 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 usb1:samplefile usb1: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 the file called samplefile from the root directory of the usb1: file system to the mystorage directory:

```
switch# copy usb1:samplefile usb1:mystorage/samplefile
```

This example shows how to copy a file from the current directory level:

```
switch# copy samplefile mystorage/samplefile
```

This example shows how to copy a file from the active supervisor module bootflash to the standby supervisor module bootflash:

```
switch# copy bootflash:nx-os-image bootflash://sup-2/nx-os-image
```

This example shows how to overwrite the contents of an existing configuration in NVRAM:

```
switch# copy nvram:snapshot-config nvram:startup-config
```

```
Warning: this command is going to overwrite your current startup-config:
Do you wish to continue? {y/n} [y] y
```

You can also use the **copy** command to upload and download files from the bootflash: file system to or from a 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.

SUMMARY STEPS

1. (Optional) **pwd**
2. (Optional) **dir** *[filesystem :[/module/][directory]]*
3. **rmdir** *[filesystem :[/module/]]directory*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) pwd Example: switch# pwd	Displays the name of your current default directory.
Step 2	(Optional) dir <i>[filesystem :[/module/][directory]]</i> 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 <i>[filesystem :[/module/]]directory</i> 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 usb1: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 2013 Samplefile
...
switch# gzip volatile:Samplefile
switch# dir
 266069      Jul 04 00:51:03 2013 Samplefile.gz
...

```

This example shows how to uncompress a compressed file:

```
switch# dir
 266069      Jul 04 00:51:03 2013 Samplefile.gz
...
switch# gunzip samplefile
switch# dir
 1525859      Jul 04 00:51:03 2013 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 > usb1: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 2013 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
```




CHAPTER 10

Working with Configuration Files

This chapter contains these sections:

- [About Configuration Files, on page 121](#)
- [Guidelines and Limitations for Configuration Files, on page 122](#)
- [Managing Configuration Files, on page 122](#)
- [Verifying the Device Configuration, on page 132](#)
- [Examples of Working with Configuration Files, on page 133](#)

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 122
- [Downloading the Startup Configuration From a Remote Server](#), on page 124

Guidelines and Limitations for Configuration Files

Configuration file guidelines and limitations are as follows:

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.

SUMMARY STEPS

1. (Optional) **show running-config**
2. **copy running-config startup-config**

DETAILED STEPS**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.

SUMMARY STEPS

1. **copy running-config** *scheme://server/[url /]filename*
2. **copy startup-config** *scheme://server/[url /]filename*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	copy running-config <i>scheme://server/[url /]filename</i> Example: <pre>switch# copy running-config tftp://10.10.1.1/sw1-run-config.bak</pre>	Copies the running-configuration file to a remote server. For the <i>scheme</i> argument, you can enter tftp: , ftp: , scp: , or sftp: . The <i>server</i> argument is the address or name of the remote server, and the <i>url</i> argument is the path to the source file on the remote server. The <i>server</i> , <i>url</i> , and <i>filename</i> arguments are case sensitive.
Step 2	copy startup-config <i>scheme://server/[url /]filename</i> Example: <pre>switch# copy startup-config tftp://10.10.1.1/sw1-start-config.bak</pre>	Copies the startup-configuration file to a remote server. For the <i>scheme</i> argument, you can enter tftp: , ftp: , scp: , or sftp: . The <i>server</i> argument is the address or name of the remote server, and the <i>url</i> argument is the path to the source file on the remote server. The <i>server</i> , <i>url</i> , and <i>filename</i> 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 device has a route to the remote server. Your 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.

SUMMARY STEPS

1. **copy** *scheme://server/[url/]filename* **running-config**

2. (Optional) **show running-config**
3. (Optional) **copy running-config startup-config**
4. (Optional) **show startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	copy <i>scheme://server/[url]/filename</i> running-config Example: <pre>switch# copy tftp://10.10.1.1/my-config running-config</pre>	Downloads the running-configuration file from a remote server. For the <i>scheme</i> argument, you can enter tftp: , ftp: , scp: , or sftp: . The <i>server</i> argument is the address or name of the remote server, and the <i>url</i> argument is the path to the source file on the remote server. The <i>server</i> , <i>url</i> , and <i>filename</i> arguments are case sensitive.
Step 2	(Optional) show running-config Example: <pre>switch# show running-config</pre>	Displays the running configuration.
Step 3	(Optional) copy running-config startup-config Example: <pre>switch# copy running-config startup-config</pre>	Copies the running configuration to the startup configuration.
Step 4	(Optional) show startup-config Example: <pre>switch# show startup-config</pre>	Displays the startup configuration.

Related Topics

[Copying Files](#), on page 116

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 device has a route to the remote server. Your 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.

SUMMARY STEPS

1. **write erase**
2. **reload**
3. **copy *scheme://server/[url /]filename* running-config**
4. **copy running-config startup-config**
5. (Optional) **show startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	write erase Example: <pre>switch# write erase</pre>	Erases the startup configuration file.
Step 2	reload Example: <pre>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#</pre>	Reloads the Cisco NX-OS device. Note Do not use the setup utility to configure the device.
Step 3	copy <i>scheme://server/[url /]filename</i> running-config Example: <pre>switch# copy tftp://10.10.1.1/my-config running-config</pre>	Downloads the running configuration file from a remote server. For the <i>scheme</i> argument, you can enter tftp: , ftp: , scp: , or sftp: . The <i>server</i> argument is the address or name of the remote server, and the <i>url</i> argument is the path to the source file on the remote server. The <i>server</i> , <i>url</i> , and <i>filename</i> arguments are case sensitive.
Step 4	copy running-config startup-config Example: <pre>switch# copy running-config startup-config</pre>	Saves the running configuration file to the startup configuration file.

	Command or Action	Purpose
Step 5	(Optional) show startup-config Example: switch# show startup-config	Displays the running configuration.

Related Topics

[Copying Files](#), on page 116

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.

SUMMARY STEPS

1. (Optional) **dir** {usb1: | usb2:}[*directory*]
2. **copy running-config** {usb1: | usb2:}[*directory*]*filename*
3. **copy startup-config** {usb1: | usb2:}[*directory*]*filename*

DETAILED STEPS**Procedure**

	Command or Action	Purpose
Step 1	(Optional) dir {usb1: usb2:}[<i>directory</i>] Example: switch# dir usb1:	Displays the files on the external flash memory device.
Step 2	copy running-config {usb1: usb2:}[<i>directory</i>] <i>filename</i> Example: switch# copy running-config usb1:dsn-running-config.cfg	Copies the running configuration to an external flash memory device. The <i>filename</i> argument is case sensitive.
Step 3	copy startup-config {usb1: usb2:}[<i>directory</i>] <i>filename</i> Example: switch# copy startup-config usb1:dsn-startup-config.cfg	Copies the startup configuration to an external flash memory device. The <i>filename</i> argument is case sensitive.

Related Topics

[Copying Files](#), on page 116

Copying the Running Configuration from an External Flash Memory Device

You can configure your 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.

SUMMARY STEPS

1. (Optional) **dir** {usb1: | usb2:}[*directory*]
2. **copy** {usb1: | usb2:}[*directory*]*filename* **running-config**
3. (Optional) **show running-config**
4. (Optional) **copy running-config startup-config**
5. (Optional) **show startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) dir {usb1: usb2:}[<i>directory</i>] Example: switch# dir usb1:	Displays the files on the external flash memory device.
Step 2	copy {usb1: usb2:}[<i>directory</i>] <i>filename</i> running-config Example: switch# copy usb1:dsn-config.cfg running-config	Copies the running configuration from an external flash memory device. The <i>filename</i> argument is case sensitive.
Step 3	(Optional) show running-config Example: switch# show running-config	Displays the running configuration.
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.

Related Topics

[Copying Files](#), on page 116

Copying the Startup Configuration From an External Flash Memory Device

You can recover the startup configuration on your 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.

SUMMARY STEPS

1. (Optional) `dir {usb1: | usb2:}[directory/]`
2. `copy {usb1: | usb2:}[directory/]filename startup-config`
3. (Optional) `show startup-config`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) <code>dir {usb1: usb2:}[directory/]</code> Example: switch# dir usb1:	Displays the files on the external flash memory device.
Step 2	<code>copy {usb1: usb2:}[directory/]filename startup-config</code> Example: switch# copy usb1:dsn-config.cfg startup-config	Copies the startup configuration from an external flash memory device. The <i>filename</i> argument is case sensitive.
Step 3	(Optional) <code>show startup-config</code> Example: switch# show startup-config	Displays the startup configuration.

Related Topics

[Copying Files](#), on page 116

Copying Configuration Files to an Internal File System

You can copy configuration files to the internal memory as a backup for later use.

SUMMARY STEPS

1. `copy running-config [filesystem:][directory/] | [directory/]filename`
2. `copy startup-config [filesystem:][directory/] | [directory/]filename`

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	copy running-config [<i>filesystem:</i>][<i>directory/</i>] [<i>directory/</i>] <i>filename</i> Example: <pre>switch# copy running-config bootflash:sw1-run-config.bak</pre>	Copies the running-configuration file to internal memory. The <i>filesystem</i> , <i>directory</i> , and <i>filename</i> arguments are case sensitive.
Step 2	copy startup-config [<i>filesystem:</i>][<i>directory/</i>] [<i>directory/</i>] <i>filename</i> Example: <pre>switch# copy startup-config bootflash:sw1-start-config.bak</pre>	Copies the startup-configuration file to internal memory. The <i>filesystem</i> , <i>directory</i> , and <i>filename</i> arguments are case sensitive.

Related Topics

[Copying Files](#), on page 107

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.

SUMMARY STEPS

1. **write erase**
2. **reload**
3. **copy configuration-file running-configuration**
4. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	write erase Example:	Clears the current configuration of the switch.

	Command or Action	Purpose
	<code>switch# write erase</code>	
Step 2	reload Example: <code>switch# reload</code>	Restarts the device. You will be prompted to provide an nx-os image file for the device to boot and run.
Step 3	copy <i>configuration-file</i> running-configuration Example: <code>switch# copy bootflash:start-config.bak running-configuration</code>	Copies a previously saved configuration file to the running configuration. Note The <i>configuration-file</i> filename argument is case sensitive.
Step 4	copy running-config startup-config Example: <code>switch# copy running-config startup-config</code>	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.

SUMMARY STEPS

1. (Optional) **show hardware**
2. **purge module *slot* running-config**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) show hardware Example: <code>switch# show hardware</code>	Displays the installed hardware for the device.
Step 2	purge module <i>slot</i> running-config Example:	Removes the configuration for a missing module from the running configuration.

	Command or Action	Purpose
	<code>switch# purge module 3 running-config</code>	
Step 3	(Optional) copy running-config startup-config Example: <code>switch# copy running-config startup-config</code>	Copies the running configuration to the startup configuration.

Erasing a Configuration

SUMMARY STEPS

1. write erase [boot | debug]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	write erase [boot debug] Example: <code>switch# write erase</code> Warning: This command will erase the startup-configuration. Do you wish to proceed anyway? (y/n) [n] y	Erases configurations in persistent memory. The default action erases the startup configuration. The boot option erases the boot variable definitions and the IPv4 configuration on the mgmt0 interface. The debug option erases the debugging configuration. Note The running-configuration file is not affected by this command.

Clearing Inactive Configurations

You can clear inactive QoS and/or ACL configurations.

SUMMARY STEPS

1. (Optional) **show running-config type inactive-if-config**
2. **clear inactive-config policy**
3. (Optional) **show inactive-if-config log**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) show running-config type inactive-if-config Example: <pre># show running-config ipqos inactive-if-config</pre>	Displays any inactive access control list (ACL) or quality of service (QoS) configurations. The values for the <i>type</i> argument are aclmgr and ipqos . <ul style="list-style-type: none"> • aclmgr—Displays any inactive configurations for aclmgr. • ipqos—Displays any inactive configurations for qosmgr.
Step 2	clear inactive-config policy Example: <pre># 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 you can see the log file @ show inactive-if-config log</pre>	Clears inactive configurations. The values for the <i>policy</i> argument are qos and acl . The following describes the values: <ul style="list-style-type: none"> • 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: <pre># show inactive-if-config log</pre>	Displays the commands that were used to clear the inactive configurations.

Verifying the Device Configuration

To verify the configuration, use one of the following commands:

Command	Purpose
show running-config	Displays the running configuration.
show startup-config	Displays the startup configuration. Note If Layer 3 based feature configurations are disabled in the running-config, the show startup-config command does not display them. However, the configurations remain intact in the startup PSS, until the copy running startup command is performed.

This option masks the sensitive words in running configuration output with <removed> keyword.

Examples of Working with Configuration Files

This section includes examples of working with configuration files.

Copying Configuration Files

This example shows how to overwrite the contents of an existing configuration in NVRAM:

```
switch# copy nvram:snapshot-config nvram:startup-config
Warning: this command is going to overwrite your current startup-config.
Do you wish to continue? {y/n} [y] y
```



Note This command does not apply to Cisco Nexus 9300-EX Series switches.

This example shows how to copy a running configuration to the bootflash: file system:

```
switch# copy system:running-config bootflash:my-config
```

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.
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.



INDEX

A

alias [61–62](#)

B

banner motd [93–94](#)

bootflash [102](#)

C

cd [104–105, 112](#)

clear inactive-config [131–132](#)

clear line [88](#)

cli alias name [61–62](#)

cli var name [59–60](#)

clock [65](#)

clock protocol none [98](#)

clock protocol ntp [98](#)

clock set [65, 97–98](#)

clock summer-time [96](#)

clock timezone [95](#)

configure [56](#)

configuring devices [23](#)

 using POAP [23](#)

copy [108, 125, 129–130](#)

copy {usb1 | usb2} [128](#)

copy ftp [124–125](#)

copy nvram [133](#)

copy running-config [128–129](#)

copy running-config {usb1 | usb2} [126](#)

copy running-config ftp [123](#)

copy running-config scp [123](#)

copy running-config sftp [123](#)

copy running-config tftp [123](#)

copy scp [124–125](#)

copy sftp [124–125](#)

copy startup-config [128–129](#)

copy startup-config {usb1 | usb2} [126](#)

copy startup-config ftp [123](#)

copy startup-config scp [123](#)

copy startup-config sftp [123](#)

copy startup-config tftp [123](#)

copy system [133](#)

copy tftp [124–125](#)

copy usb1 | usb2 [127](#)

D

databits [84–85](#)

debug [102](#)

diff-clean [69](#)

diff-clean all-sessions [69](#)

diff-clean all-users [69](#)

dir [107–108, 110](#)

dir usb1 [126–128](#)

dir usb2 [126–128](#)

E

echo [64](#)

echo backslash-interpret [64](#)

egrep [70](#)

email [77](#)

end [51](#)

exec-timeout [84–86](#)

F

find [112](#)

from [77–78](#)

G

grep [70](#)

gunzip [110](#)

gzip [110](#)

H

hostname [92–93](#)

I

interface [49–50](#)

ip {ftp | http | tftp} source-interface {ethernet | loopback} [103](#)

L

less [70](#)
 line console [84–85](#)
 line vty [86–87](#)
 log [102](#)

M

mkdir [105](#)
 move [107](#)

P

parity {even | none | odd} [84–85](#)
 persist [75](#)
 ping [123, 125](#)
 ping6 [123, 125](#)
 POAP [23, 31](#)
 setting up the network environment [31](#)
 pop [50](#)
 provisioning devices [23](#)
 using POAP [23](#)
 purge module [130](#)
 push [50](#)
 pwd [104–105, 108, 111–112](#)

R

reload [125, 129–130, 133](#)
 reply-to [77–78](#)
 rmdir [106, 117](#)
 run-script [63](#)

S

sed [71](#)
 send [99](#)
 session-limit [87](#)
 set [65](#)
 show [66–67, 69, 71–72, 79](#)
 show banner motd [93–94](#)
 show cli history [73–74](#)
 show cli variables [59–60](#)
 show clock [95, 97](#)
 show clock detail [96](#)
 show email [77–78](#)
 show file [109–110](#)
 show hardware [130](#)

show inactive-if-config log [131–132](#)
 show interface brief [81](#)
 show line [89](#)
 show line console [85–86](#)
 show run clock_manager [98](#)
 show running-config [32, 99, 122, 124, 127, 131–132](#)
 show startup-config [32, 99, 125–128, 132](#)
 show terminal [89](#)
 show users [88–89, 98–99](#)
 sleep [64](#)
 slot [76](#)
 smtp-host [77](#)
 smtp-port [77](#)
 sort [71](#)
 speed [85](#)
 ssh name [69](#)
 stopbits [85](#)
 switchname [92–93](#)
 system [102](#)

T

tail [111](#)
 tar append bootflash [114](#)
 tar append volatile [114](#)
 tar create bootflash [113](#)
 tar create volatile [113](#)
 tar extract {bootflash | volatile} [115](#)
 tar list bootflash [115](#)
 tar list volatile [115](#)
 terminal alias [62](#)
 terminal color [75](#)
 terminal dont-ask [75](#)
 terminal edit-mode vi [74](#)
 terminal output xml [69](#)
 terminal redirection-mode ascii [111](#)
 terminal redirection-mode zipped [111](#)

V

volatile [102](#)
 vrf management [77–78](#)

W

where detail [57](#)
 write erase [125, 129, 133](#)
 write erase boot [131](#)
 write erase debug [131](#)