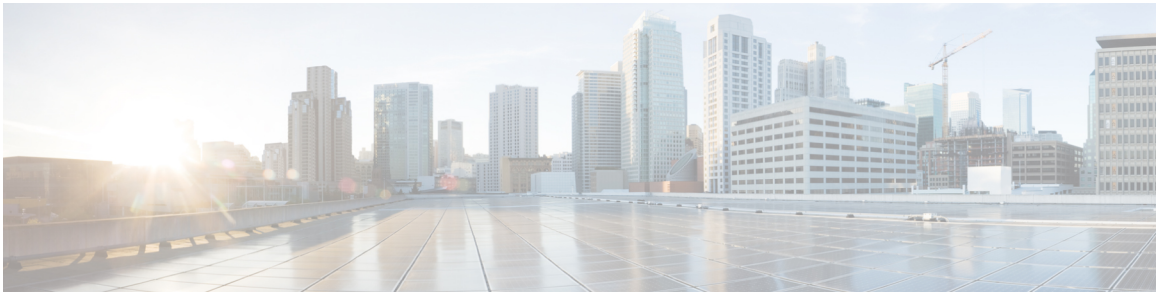


Cisco Nexus 3000 Series NX-OS Fundamentals Configuration Guide, Release 7.x

First Published: 2015-08-18

Last Modified: 2019-03-20

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CONTENTS

PREFACE

Preface xi

Audience xi

Document Conventions xi

Related Documentation for Cisco Nexus 3000 Series Switches xii

Documentation Feedback xii

Communications, Services, and Additional Information xii

CHAPTER 1

New and Changed Information 1

New and Changed Information for this Release 1

CHAPTER 2

Understanding the Command-Line Interface 3

Licensing Requirements 3

Information About the CLI Prompt 4

Command Modes 4

EXEC Command Mode 4

Global Configuration Command Mode 5

Interface Configuration Command Mode 5

Subinterface Configuration Command Mode 6

Saving and Restoring a Command Mode 6

Exiting a Configuration Command Mode 6

Command Mode Summary 7

Special Characters 8

Keystroke Shortcuts 8

Abbreviating Commands 10

Completing a Partial Command Name 11

Identifying Your Location in the Command Hierarchy 11
Using the no Form of a Command 12
Configuring CLI Variables 13
About CLI Variables 13
Configuring CLI Session-Only Variables 14
Configuring Persistent CLI Variables 14
Command Aliases 15
About Command Aliases 15
Defining Command Aliases 16
Configuring Command Aliases for a User Session 16
Command Scripts 17
Running a Command Script 17
Echoing Information to the Terminal 17
Delaying Command Action 18
Context-Sensitive Help 18
Understanding Regular Expressions 19
Special Characters 20
Multiple-Character Patterns 20
Anchoring 20
Searching and Filtering show Command Output 21
Filtering and Searching Keywords 22
diff Utility 23
grep and egrep Utilities 24
less Utility 25
sed Utility 25
sort Utility 25
Searching and Filtering from the --More-- Prompt 26
Using the Command History 27
Recalling a Command 27
Controlling CLI History Recall 27
Configuring the CLI Edit Mode 27
Displaying the Command History 27
Enabling or Disabling the CLI Confirmation Prompts 28
Setting CLI Display Colors 28

Sending Commands to Modules 29
BIOS Loader Prompt 29
Examples Using the CLI 30
Defining Command Aliases 30
Using CLI Session Variables 30
Using the System-Defined Timestamp Variable 31
Running a Command Script 31
Additional References for the CLI 32
Related Documents for the CLI 32

CHAPTER 3 **Configuring Terminal Settings and Sessions** 33
Information About Terminal Settings and Sessions 33
Terminal Session Settings 33
Console Port 33
Virtual Terminals 34
Modem Support 34
Configuring the Console Port 35
Configuring Virtual Terminals 36
Configuring the Inactive Session Timeout 36
Configuring the Session Limit 37
Configuring Modem Connections 38
Enabling a Modem Connection 38
Downloading the Default Initialization String 38
Configuring and Downloading a User-Specified Initialization String 39
Initializing a Modem for a Powered-Up Cisco NX-OS Device 40
Clearing Terminal Sessions 40
Displaying Terminal and Session Information 41
Default Settings for File System Parameters 41
Additional References for Terminal Settings and Sessions 41
Related Documents for Terminal Settings and Sessions 42

CHAPTER 4 **Basic Device Management** 43
Information About Basic Device Management 43
Device Hostname 43

Message-of-the-Day Banner 43
Device Clock 43
Clock Manager 44
Time Zone and Summer Time (Daylight Saving Time) 44
User Sessions 44
Changing the Device Hostname 44
Configuring the MOTD Banner 45
Configuring the Time Zone 46
Configuring Summer Time (Daylight Saving Time) 46
Manually Setting the Device Clock 47
Setting the Clock Manager 48
Configuring the Mode on the Cisco Nexus 3100 Series Switches 49
Managing Users 54
Displaying Information about the User Sessions 54
Sending a Message to Users 54
Verifying the Device Configuration 55
Default Settings for Basic Device Parameters 55
Additional References for Basic Device Management 55
Related Documents for Basic Device Management 55

CHAPTER 5 **Using Power On Auto Provisioning 57**
Information About PowerOn Auto Provisioning 57
Network Requirements for POAP 57
POAP Configuration Script 58
Using the POAP Script and POAP Script Options 58
POAP Upgrade Path 62
POAP Upgrade Bootflash Size 62
Setting up the DHCP Server without DNS for POAP 63
Downloading and Using User Data, Agents, and Scripts as part of POAP 63
Troubleshooting for POAP 64
POAP Process 64
Power-Up Phase 65
USB Discovery Phase 66
DHCP Discovery Phase 66

POAP Dynamic Breakout 67
Script Execution Phase 67
Post-Installation Reload Phase 68
Guidelines and Limitations for POAP 68
Setting Up the Network Environment To Use POAP 69
Configuring a Switch Using POAP 70
Verifying the Device Configuration 71
Related Documents for POAP 71

CHAPTER 6 **Using the Device File Systems, Directories, and Files 73**
Information About the Device File Systems, Directories, and Files 73
File Systems 73
Directories 74
Files 74
Working with Directories 74
Identifying the Current Directory 74
Changing the Current Directory 75
Creating a Directory 75
Displaying Directory Contents 75
Deleting a Directory 76
Accessing Directories on the Standby Supervisor Module 76
Working with Files 76
Moving Files 77
Copying Files 77
Configuring the Source Interface for Copying Files to or from a Remote Server 78
Copying Files to the HTTP Server 79
Deleting Files 80
Displaying File Contents 81
Displaying File Checksums 81
Compressing and Uncompressing Files 81
Displaying the Last Lines in a File 82
Redirecting show Command Output to a File 82
Finding Files 83
Working with Archive Files 83

Creating an Archive Files 83
Appending Files to an Archive File 84
Extracting Files from an Archive File 85
Displaying the Filenames in an Archive File 86
Examples of Using the File System 86
Accessing Directories on Standby Supervisor Modules 86
Moving Files 86
Copying Files 87
Deleting a Directory 87
Displaying File Contents 88
Displaying File Checksums 88
Compressing and Uncompressing Files 88
Redirecting show Command Output 89
Finding Files 89
Default Settings for File System Parameters 89
Additional References for File Systems 90
Related Documents for File Systems 90

CHAPTER 7 **Working with Configuration Files** 91
Information About Configuration Files 91
Types of Configuration Files 91
Guidelines and Limitations for Configuration Files 92
Managing Configuration Files 92
Copying Configuration Files to the Startup Configuration 92
Configuring the Source Interface for Copying Configuration Files to or from a Remote Server 92
Saving the Running Configuration to the Startup Configuration 93
Copying a Configuration File to a Remote Server 93
Downloading the Running Configuration From a Remote Server 94
Downloading the Startup Configuration From a Remote Server 95
Copying Configuration Files to an External Flash Memory Device 97
Copying the Running Configuration from an External Flash Memory Device 98
Copying the Startup Configuration from an External Flash Memory Device 99
Copying Configuration Files to an Internal File System 99

Rolling Back to a Previous Configuration 100
Erasing a Configuration 101
Clearing Inactive Configurations 102
Configuration Archive and Configuration Log 103
Information About Configuration Archive 103
Configuring the Characteristics of the Configuration Archive 103
Information About Configuration Log 105
Displaying Configuration Log Entries 106
Verifying the Device Configuration 107
Examples of Working with Configuration Files 107
Copying Configuration Files 107
Backing Up Configuration Files 107
Rolling Back to a Previous Configuration 107
Additional References for Configuration Files 108
Related Documents for Configuration Files 108

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

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

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

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

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

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

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: http://www.cisco.com/go/trademarks. 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. (1110R)

© 2017–2019 Cisco Systems, Inc. All rights reserved.



Preface

This preface includes the following sections:

- Audience, on page xi
- Document Conventions, on page xi
- Related Documentation for Cisco Nexus 3000 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.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

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

Examples use the following conventions:

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

Related Documentation for Cisco Nexus 3000 Series Switches

The entire Cisco Nexus 3000 Series switch documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/tsd-products-support-series-home.html

Documentation Feedback

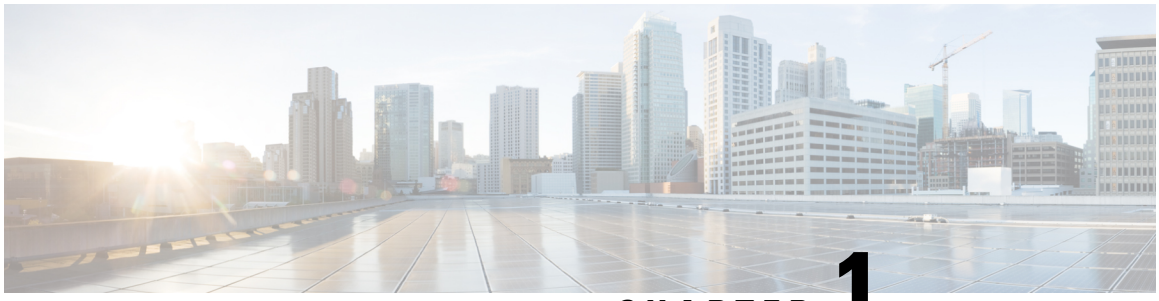
To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3k-docfeedback@cisco.com. We appreciate your feedback.

Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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



CHAPTER 1

New and Changed Information

This chapter contains the following sections:

- New and Changed Information for this Release, on page 1

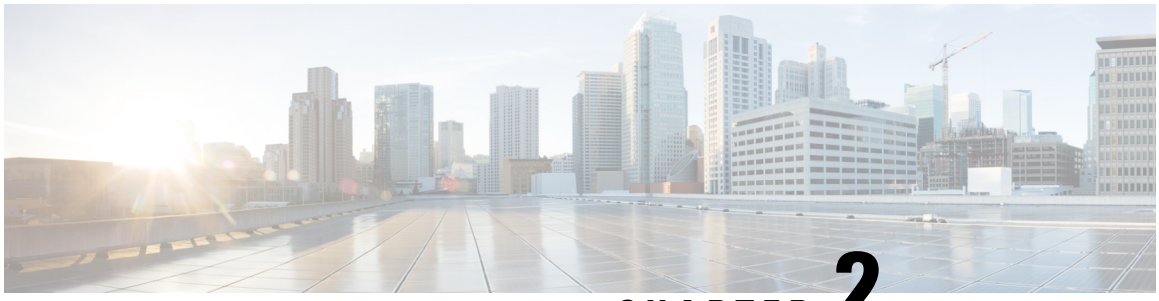
New and Changed Information for this Release

The following table provides an overview of the significant changes to this guide for this current release. The table does not provide an exhaustive list of all changes made to the configuration guides or of the new features in this release.

Table 1: New and Changed Features

Feature	Description	Added or Changed in Release	Where Documented
Reload timer	Added support for **reload timer** command.	7.0(3)I7(4)	-
POAP	Added support for RFC 3004 (User Class Option for DHCP).	7.0(3)I7(4)	Guidelines and Limitations for POAP, on page 68
POAP	Added support for POAP over IPv6.	7.0(3)I7(4)	Guidelines and Limitations for POAP, on page 68
Configuration Archive and Configuration Log	New sections added for Configuration Archive and Configuration Log.	7.0(3)I5(1)	Configuration Archive and Configuration Log, on page 103
Command-line interface	Enabled Cisco NX-OS CLI variables to contain hyphens and underscores.	7.0(3)I4(1)	Configuring CLI Session-Only Variables, on page 14 Configuring Persistent CLI Variables, on page 14

Feature	Description	Added or Changed in Release	Where Documented
POAP	Introduced the POAP personality and dynamic breakout features.	7.0(3)I4(1)	Information About PowerOn Auto Provisioning, on page 57
System Mode	Describes the two system modes, the N3K mode and the N9K mode supported on the Cisco Nexus 3100 Series switches.	7.0(3)I2(1)	Configuring the Mode on the Cisco Nexus 3100 Series Switches , on page 49
Ability to support HTTPS for downloading the files.	Added support for the HTTPS protocol for downloading the files.	7.0(3)I2(1)	Downloading the Startup Configuration From a Remote Server, on page 95 Downloading the Running Configuration From a Remote Server Copying a Configuration File to a Remote Server, on page 93



CHAPTER 2

Understanding the Command-Line Interface

This chapter contains the following sections:

- Licensing Requirements, on page 3
- Information About the CLI Prompt, on page 4
- Command Modes, on page 4
- Special Characters, on page 8
- Keystroke Shortcuts, on page 8
- Abbreviating Commands, on page 10
- Completing a Partial Command Name, on page 11
- Identifying Your Location in the Command Hierarchy, on page 11
- Using the no Form of a Command , on page 12
- Configuring CLI Variables, on page 13
- Command Aliases, on page 15
- Command Scripts, on page 17
- Context-Sensitive Help , on page 18
- Understanding Regular Expressions, on page 19
- Searching and Filtering show Command Output, on page 21
- Searching and Filtering from the --More-- Prompt, on page 26
- Using the Command History, on page 27
- Enabling or Disabling the CLI Confirmation Prompts, on page 28
- Setting CLI Display Colors, on page 28
- Sending Commands to Modules, on page 29
- BIOS Loader Prompt, on page 29
- Examples Using the CLI , on page 30
- Additional References for the CLI, on page 32

Licensing Requirements

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

Information About the CLI Prompt

Once you have successfully accessed the device, the CLI prompt displays in the terminal window of your console port or remote workstation as shown in 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) 2002-2009, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
switch#
```

You can change the default device hostname.

From the CLI prompt, you can do the following:

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



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

Command Modes

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

EXEC Command Mode

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

Global Configuration Command Mode

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

Procedure

	Command or Action	Purpose

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

Interface Configuration Command Mode

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

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

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

Procedure

	Command or Action	Purpose

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

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

Subinterface Configuration Command Mode

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

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

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

Procedure

	Command or Action	Purpose

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

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

Saving and Restoring a Command Mode

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

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

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

Exiting a Configuration Command Mode

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

Procedure

	Command or Action	Purpose

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

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

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

Command Mode Summary

This table summarizes information about the main command modes.

Table 2: Command Mode Summary

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

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

Special Characters

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

Table 3: 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 4: Keystroke Shortcuts

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

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

Keystokes	Description

Tab	Completes the word for you after you enter the first characters of the word and then press the Tab key. All options that match are presented.
	Use tabs to complete the following items:
	• Command names
	• Scheme names in the file system
	• Server names in the file system
	• Filenames in the file system
	Example:
	`switch(config)# xm<Tab>`
	`switch(config)# xml<Tab>`
	`switch(config)# xml server`

	Example:
	`switch(config)# c<Tab>`
	`callhome  class-map  clock  cts`
	`cdp  cli  control-plane`
	`switch(config)# cl<Tab>`
	`class-map  cli  clock`
	`switch(config)# cla<Tab>`
	`switch(config)# class-map`

	Example:
	`switch# cd bootflash:<Tab>`
	`bootflash:  bootflash://sup-1/`
	`bootflash:///  bootflash://sup-2/`
	`bootflash://module-5/  bootflash://sup-active/`
	`bootflash://module-6/  bootflash://sup-local/`

	Example:
	`switch# cd bootflash://mo<Tab>`
	`bootflash://module-5/  bootflash://module-6/cv`
	`switch# cd bootflash://module-`

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 5: 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 submode hierarchy nested more than one level. In these cases, you can display information about your present working context (PWC).

Procedure

	Command or Action	Purpose

Step 1	**where detail**
	Example:
	`switch# configure terminal`
	`switch(config)# interface mgmt0`
	`switch(config-if)# where detail`
	`mode:  conf`
	`interface mgmt0`
	`username:  admin`
		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. The Cisco NX-OS command reference publications describe the function of the **no** form of the command whenever a **no** form is available.

This example shows how to disable a feature:

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

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

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

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

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

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

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

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

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

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

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

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

Configuring CLI Variables

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

About CLI Variables

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

You can refer to CLI variables in the following ways:

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

CLI variables have the following characteristics:

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

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



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

Configuring CLI Session-Only Variables

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

Procedure

	Command or Action	Purpose
Step 1	**cli var name** *variable-name variable-text* **Example:** `switch# cli var name testinterface ethernet 2/1`	Configures the CLI session variable. The *variable-name* argument is alphanumeric, case sensitive, and has a maximum length of 31 characters. The *variable-text* argument is alphanumeric, case sensitive, can contain spaces, and has a maximum length of 200 characters. **Note** Beginning with Cisco NX-OS Release 7.0(3)I4(1), variables can include hyphens (-) and underscores (_).
Step 2	(Optional) **show cli variables** **Example:** `switch# show cli variables`	Displays the CLI variable configuration.

Configuring Persistent CLI Variables

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

Procedure

	Command or Action	Purpose

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

Step 2	**cli var name** *variable-name variable-text* **Example:** `switch(config)# cli var name testinterface ethernet 2/1`	Configures the CLI persistent variable. The variable name is a case-sensitive, alphanumeric string and must begin with an alphabetic character. The maximum length is 31 characters. **Note** Beginning with Cisco NX-OS Release 7.0(3)I4(1), variables can include hyphens (-) and underscores (_).

	Command or Action	Purpose

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

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

Command Aliases

This section provides information about command aliases.

About Command Aliases

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

Command alias support has the following characteristics:

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

Defining Command Aliases

You can define command aliases for commonly used commands.

Procedure

	Command or Action	Purpose

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

Step 2	**cli alias name** *alias-name alias-text* **Example:** `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.

Procedure

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

Command Scripts

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

Running a Command Script

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

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

Procedure

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

Echoing Information to the Terminal

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

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

Table 6: Formatting Options for the echo Command

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

Formatting Option	Description
\nnn	Displays the corresponding ASCII octal character.

Procedure

	Command or Action	Purpose

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

Delaying Command Action

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

Procedure

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

Context-Sensitive Help

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

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

This table shows example outputs of context sensitive help.

Table 7: Context-Sensitive Help Example

Example Outputs	Description

`switch# clock ?`
`  set  HH:MM:SS Current Time`
`switch# clock`
	Displays the command syntax for the **clock** command in EXEC mode.
	The switch output shows that the **set** keyword is required for using the **clock** command.

Example Outputs	Description

`switch# clock set ?`
`  WORD  HH:MM:SS Current Time`
`switch# clock set`
	Displays the command syntax for setting the time.
	The help output shows that the current time is required for setting the clock and how to format the time.

`switch# clock set 13:32:00<CR>`
`% Incomplete command`
`switch#`
	Adds the current time.
	The CLI indicates the command is incomplete.

`switch# <Ctrl-P>` `switch# clock set 13:32:00`	Displays the previous command that you entered.

`switch# clock set 13:32:00 ?`
`  <1-31>  Day of the month`
`switch# clock set 13:32:00`
	Displays the additional arguments for the **clock set** command.

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

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

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

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

Understanding Regular Expressions

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

Special Characters

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

This table lists the keyboard characters that have special meanings.

Table 8: 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 9: Special Characters Used for Anchoring

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

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



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

Searching and Filtering show Command Output

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

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

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

Keyword Syntax	Description
human **Example:** `show version | human`	Displays the output in normal format if you have previously set the output format to XML using the **terminal output xml** command. **Note** The **terminal output xml** command, having 1.0 and 6.x / 7.x versions is used to set the schema (.xsd) file while validating xml. Setting 1.0, the switch will validate /isan/etc/schema/<comp>.xsd file, and if 6.x / 7.x the switch validates /isan/ect/schema/6.x/<comp>.xsd file and /isan/ect/schema/7.x/<comp>.xsd file accordingly based on the version of the image, the switch is running.
include *string* **Example:** `show interface brief | include up`	Displays all lines that include the search string. The search string is case sensitive.
last [*lines*] **Example:** `show logging logfile | last 50`	Displays the end of the output for the number of lines specified. The default number of lines is 10.
no-more **Example:** `show interface brief | no-more`	Displays all the output without stopping at the end of the screen with the `--More--` prompt.
sscp *SSH-connection-name filename* **Example:** `show version | sscp MyConnection show_version_output`	Redirects the output using streaming secure copy (sscp) to a named SSH connection. You can create the SSH named connection using the **ssh name** command.
wc [**bytes** | **lines** | **words**] **Example:** `show file testoutput | wc bytes`	Displays counts of characters, lines, or words. The default is to display the number of lines, words, and characters.
xml **Example:** `show version | xml`	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-session**] [**all-users**]

This table describes the keywords for the diff utility.

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

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

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

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

grep and egrep Utilities

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

The grep and egrep syntax is as follows:

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

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

Table 11: grep and egrep Parameters

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

less Utility

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

sed Utility

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

sed *command*

The *command* argument contains sed utility commands.

sort Utility

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

The sort utility syntax is as follows:

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

This table describes the sort utility parameters.

Table 12: 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 *field-number*[**.***char-position*][*ordering*]	Sorts according to a key value. There is no default key value.
-n	Sorts according to a numeric string value.
-r	Reverses order of the sort results. The default sort output is in ascending order.
-t *delimiter*	Sorts using a specified delimiter. The default delimiter is the space character.
-u	Removes duplicate lines from the sort results. The sort output displays the duplicate lines.

Searching and Filtering from the --More-- Prompt

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

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

Table 13: --More-- Prompt Commands

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

Using the Command History

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

Recalling a Command

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

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

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

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

Controlling CLI History Recall

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

Configuring the CLI Edit Mode

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

Procedure

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

Displaying the Command History

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

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

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

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

```
switch# show cli history
```

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

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

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

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

Enabling or Disabling the CLI Confirmation Prompts

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

Procedure

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

Setting CLI Display Colors

You can change the CLI colors to display as follows:

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

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

Procedure

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

Sending Commands to Modules

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

The **slot** has the following syntax:

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

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

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

The following example shows how to display and filter module information:

```
switch# slot 1 show version | grep lc
```

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

```
switch# slot 1 quoted "show version" | diff
```

BIOS Loader Prompt

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

```
loader>
```

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

Examples Using the CLI

This section includes examples of using the CLI.

Defining Command Aliases

This example shows how to define command aliases:

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

This example shows how to use a command alias:

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

Using CLI Session Variables

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

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

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

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

Using the System-Defined Timestamp Variable

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

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

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

Running a Command Script

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

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

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

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

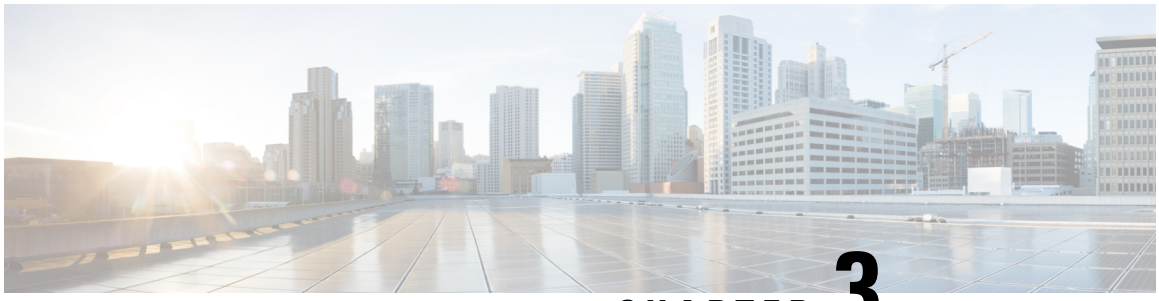
```
  0 broadcast packets 0 jumbo packets 0 storm suppression packets
  0 bytes
Tx
  0 output packets 0 multicast packets
  0 broadcast packets 0 jumbo packets
  0 bytes
  0 input error 0 short frame 0 watchdog
  0 no buffer 0 runt 0 CRC 0 ecc
  0 overrun  0 underrun 0 ignored 0 bad etype drop
  0 bad proto drop 0 if down drop 0 input with dribble
  0 input discard
  0 output error 0 collision 0 deferred
  0 late collision 0 lost carrier 0 no carrier
  0 babble
  0 Rx pause 0 Tx pause 0 reset
```

Additional References for the CLI

This section includes additional information related to the CLI.

Related Documents for the CLI

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 3000 Series NX-OS Command Reference*



CHAPTER 3

Configuring Terminal Settings and Sessions

This chapter contains the following sections:

- Information About Terminal Settings and Sessions, on page 33
- Configuring the Console Port, on page 35
- Configuring Virtual Terminals , on page 36
- Configuring Modem Connections, on page 38
- Clearing Terminal Sessions, on page 40
- Displaying Terminal and Session Information, on page 41
- Default Settings for File System Parameters, on page 41
- Additional References for Terminal Settings and Sessions, on page 41

Information About Terminal Settings and Sessions

This section includes information about terminal settings and sessions.

Terminal Session Settings

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

Terminal type
: Name used by Telnet when communicating with remote hosts

Length
: Number of lines of command output displayed before pausing

Width
: Number of characters displayed before wrapping the line

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

Console Port

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

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

Inactive session timeout
Specifies the number of minutes a session can be inactive before it is terminated.
Parity
Specifies the odd or even parity for error detection.
Speed
Specifies the transmission speed for the connection.
Stop bits
Specifies the stop bits for an asynchronous line.

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

Virtual Terminals

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

Modem Support

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

- MultiTech MT2834BA (http://www.multitech.com/en_us/support/families/multimodemii/)
- Hayes Accura V.92 (http://www.zoom.com/products/dial_up_external_serial.html#hayes)

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

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

AT
Attention
E0 (required)
No echo
Q1
Result code on
&D2
Normal data terminal ready (DTR) option
&C1
Enable tracking the state of the data carrier
S0=1
Pick up after one ring
\015 (required)
Carriage return in octal

Configuring the Console Port

You can set the following characteristics for the console port:

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

Before you begin

Log in to the console port.

Procedure

	Command or Action	Purpose

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

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

Step 3	**databits** *bits* **Example:** `switch(config-console)# databits 7`	Configures the number of data bits per byte. The range is from 5 to 8. The default is 8.
Step 4	**exec-timeout** *minutes* **Example:** `switch(config-console)# exec-timeout 30`	Configures the timeout for an inactive session. The range is from 0 to 525600 minutes (8760 hours). A value of 0 minutes disables the session timeout. The default is 30 minutes.
Step 5	**parity** {**even** | **none** | **odd**} **Example:** `switch(config-console)# parity even`	Configures the parity. The default is **none**.
Step 6	**speed** {**300** | **1200** | **2400** | **4800** | **9600** | **38400** | **57600** | **115200**} **Example:** `switch(config-console)# speed 115200`	Configures the transmit and receive speed. The default is **115200**.

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

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

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

Configuring Virtual Terminals

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

Configuring the Inactive Session Timeout

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

Procedure

	Command or Action	Purpose

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

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

Step 3	**exec-timeout** *minutes* **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:
	`switch(config-line)# exit`
	`switch(config)#`
		Exits line configuration mode.

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

Configuring the Session Limit

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

Procedure

	Command or Action	Purpose

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

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

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

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

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

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

Configuring Modem Connections

You can connect a modem to the console port.

Enabling a Modem Connection

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

Before you begin

Log in to the console port.

Procedure

	Command or Action	Purpose

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

Step 2	**line console**	Enters console configuration mode.
Step 3	**modem in**	Enables modem input on the port.
Step 4	**exit**	Exits console configuration mode.
Step 5	(Optional) **show line** **Example:** `switch(config)# show line`	Displays the console settings.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Downloading the Default Initialization String

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

Before you begin

Log in to the console port.

Procedure

	Command or Action	Purpose

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

Step 2	**line console**	
Step 3	**modem init-string default**	Writes the default initialization string to the modem.
Step 4	**exit**	Exits console configuration mode.
Step 5	(Optional) **show line** **Example:** `switch(config)# show line`	Displays the console settings.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring and Downloading a User-Specified Initialization String

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

Before you begin

Log in to the console port.

Procedure

	Command or Action	Purpose

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

Step 2	**line console**	
Step 3	**modem set-string user-input** *string*	Sets the user-specified initialization string for the console port. The initialization string is alphanumeric and case sensitive, can contain

	Command or Action	Purpose
		special characters, and has a maximum of 100 characters. **Note** You must first set the user-input string before initializing the string.
Step 4	**modem init-string user-input**	Writes the user-specified initialization string to the modem connected to the console port.
Step 5	**exit**	Exits console configuration mode.
Step 6	(Optional) **show line** **Example:** `switch(config)# show line`	Displays the console settings.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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

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

Before you begin

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

Enable the modem connection on the port.

Procedure

	Command or Action	Purpose
Step 1	**modem connect line console**} **Example:** `switch# modem connect line console`	Initializes the modem connected to the device.

Clearing Terminal Sessions

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

Procedure

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

Displaying Terminal and Session Information

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

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

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

Default Settings for File System Parameters

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

Table 14: Default File System Settings

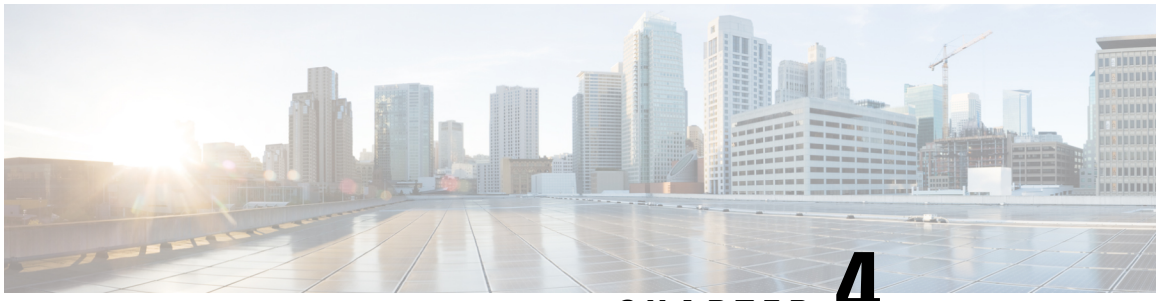
Parameters	Default
Default filesystem	bootflash:

Additional References for Terminal Settings and Sessions

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

Related Documents for Terminal Settings and Sessions

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 3000 Command Reference*



CHAPTER 4

Basic Device Management

This chapter contains the following sections:

- Information About Basic Device Management, on page 43
- Changing the Device Hostname, on page 44
- Configuring the MOTD Banner, on page 45
- Configuring the Time Zone, on page 46
- Configuring Summer Time (Daylight Saving Time), on page 46
- Manually Setting the Device Clock, on page 47
- Setting the Clock Manager, on page 48
- Configuring the Mode on the Cisco Nexus 3100 Series Switches , on page 49
- Managing Users, on page 54
- Verifying the Device Configuration, on page 55
- Default Settings for Basic Device Parameters, on page 55
- Additional References for Basic Device Management, on page 55

Information About Basic Device Management

This section provides information about basic device management.

Device Hostname

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

Message-of-the-Day Banner

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

Device Clock

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

Clock Manager

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

The clock manager provides a way to synchronize these different clocks.

Time Zone and Summer Time (Daylight Saving Time)

You can configure the time zone and summer time (daylight saving time) setting for your device. These values offset the clock time from Coordinated Universal Time (UTC). UTC is International Atomic Time (TAI) with leap seconds added periodically to compensate for the Earth's slowing rotation. UTC was formerly called Greenwich Mean Time (GMT).

User Sessions

You can display the active user session on your device. You can also send messages to the user sessions. For more information about managing user sessions and accounts, see the Cisco Nexus security configuration guide for your device.

Changing the Device Hostname

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

Procedure

	Command or Action	Purpose

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

Step 2	{**hostname** | **switchname**} *name*
	Example:
	Using the **hostname** command:
	`switch(config)# hostname Engineering1`
	`Engineering1(config)#`
	Using the **switchname** command:
	`Engineering1(config)# switchname Engineering2`
	`Engineering2(config)#`
		Changes the device hostname. The *name* argument is alphanumeric, case sensitive, and has a maximum length of 32 characters. The default is switch.
		Note The **switchname** command performs the same function as the **hostname** command.

Step 3	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`Engineering2(config)# exit` `Engineering2#`	
Step 4	(Optional) **copy running-config startup-config** **Example:** `Engineering2# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the MOTD Banner

You can configure the MOTD to display before the login prompt on the terminal when a user logs in. The MOTD banner has the following characteristics:

- Maximum of 80 characters per line
- Maximum of 40 lines

Procedure

	Command or Action	Purpose

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

Step 2	**banner motd** *delimiting-character message delimiting-character*
	Example:
	`switch(config)# banner motd #Welcome to the Switch#`
	`switch(config)#`
		Configures the MOTD banner. Do not use the *delimiting-character* in the *message* text.
		Note Do not use " or % as a delimiting character.

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

Step 4	(Optional) **show banner motd** **Example:** `switch# show banner motd`	Displays the configured MOTD banner.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Time Zone

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

Procedure

	Command or Action	Purpose

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

Step 2	**clock timezone** *zone-name offset-hours offset-minutes* **Example:** `switch(config)# clock timezone EST -5 0`	Configures the time zone. The *zone-name* argument is a 3-character string for the time zone acronym (for example, PST or EST). The *offset-hours* argument is the offset from the UTC and the range is from –23 to 23 hours. The range for the *offset-minutes* argument is from 0 to 59 minutes.

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

Step 4	(Optional) **show clock** **Example:** `switch# show clock`	Displays the time and time zone.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Summer Time (Daylight Saving Time)

You can configure when summer time, or daylight saving time, is in effect for the device and the offset in minutes.

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
Step 2	**clock summer-time** *zone-name start-week start-day start-month start-time end-week end-day end-month end-time offset-minutes* **Example:** `switch(config)# clock summer-time PDT 1 Sunday March 02:00 1 Sunday November 02:00 60`	Configures summer time or daylight saving time. The *zone-name* argument is a three character string for the time zone acronym (for example, PST and EST). The values for the *start-day* and *end-day* arguments are **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, **Saturday**, and **Sunday**. The values for the *start-month* and *end-month* arguments are **January**, **February**, **March**, **April**, **May**, **June**, **July**, **August**, **September**, **October**, **November**, and **December**. The value for the *start-time* and *end-time* arguments are in the format *hh***:***mm*. The range for the *offset-minutes* argument is from 0 to 1440 minutes.

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

Step 4	(Optional) **show clock detail** **Example:** `switch(config)# show clock detail`	Displays the configured MOTD banner.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Manually Setting the Device Clock

You can set the clock manually if your device cannot access a remote time source.

Before you begin

Configure the time zone.

Procedure

	Command or Action	Purpose

Step 1	**clock set** *time day month year*
	Example:
	`switch# clock set 15:00:00 30 May 2008`
	`Fri May 30 15:14:00 PDT 2008`
		Configures the device clock.
		The format for the *time* argument is *hh*:*mm*:*ss*.
		The range for the *day* argument is from 1 to 31.
		The values for the *month* argument are **January**, **February**, **March**, **April**, **May**, **June**, **July**, **August**, **September**, **October**, **November**, and **December**.
		The range for the *year* argument is from 2000 to 2030.

Step 2	(Optional) **show clock** **Example:** `switch(config)# show clock`	Displays the current clock value.

Related Topics

Configuring the Time Zone, on page 46

Setting the Clock Manager

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

Procedure

	Command or Action	Purpose
Step 1	**clock protocol** *protocol* **vdc** *vdc-num* **Example:** `# clock protocol ptp vdc 2`	Configures the clock manager. The values for the *protocol* argument are **ptp**, **ntp**, and **none**. The following describes the values: • **ptp**—Synchronizes clocks with Precision Time Protocol (PTP) as described by IEEE 1588. • **ntp**— Synchronizes clocks with Network Time Protocol (NTP). • **none**—Use **clock set** to set supervisor clocks. **Note** When **none** is used, the clock in the specified VDC must be configured.

	Command or Action	Purpose
		Note Once the protocol is configured, the clock in the specified VDC must use that protocol. For example, if the **clock protocol ptp vdc 2** command is entered, then PTP should be configured in VDC 2. The range for the *vdc* argument is 1 to 8.
Step 2	(Optional) **show run clock_manager** **Example:** `#show run clock_manager`	Displays the configuration of the clock manager.

Configuring the Mode on the Cisco Nexus 3100 Series Switches

You can configure the Cisco Nexus 3100 Series switches in the N9K mode using the following commands:

Before you begin

The Cisco Nexus 3100 Series switches, except Cisco Nexus 3100-V switches, now support two system modes: the N3K mode and the N9K mode. The N3K mode is the default mode. It uses the same CLI commands as the previous Cisco Nexus 3000 Series and Cisco Nexus 3100 Series NX-OS releases. The N9K mode enables the Cisco Nexus 3100 Series switches to use the Cisco Nexus 9000 Series switches CLI commands. Refer to the Cisco Nexus 9000 Series configuration guides for the Cisco Nexus 9000 Series CLI commands.



Note The N9K mode is available on the Cisco Nexus 3100 Series switches only and it is not available on the Cisco Nexus 3000 Series switches. Cisco Nexus 3100-V switches supports only N9K CLI.

Procedure

	Command or Action	Purpose

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

Step 2	switch(config)# **system switch-mode** *mode*
	Example:
	`switch(config)# system switch-mode n9k`
	`!WARNING: "write erase/reload" is required before new mode is effective.`
		Configures the mode as N9K.

	Command or Action	Purpose

Step 3	switch(config)# **write erase**
	Example:
	`switch(config)# write erase`
	`Warning: This command will erase the startup-configuration.`
	`Do you wish to proceed anyway? (y/n) [n] y`
		Erases the start-up configuration.

Step 4	switch(config)# **reload**
	Example:
	`switch(config)# reload`
	`This command will reboot the system. (y/n)?  [n] y`
	`2002 Jan  9 03:57:59 Neptune-1 %$ VDC-1 %$ %PLATFORM-2-PFM_SYSTEM_RESET: Manual system restart from Command Line Interface`
	
	`(c) Copyright 2013, Cisco Systems.`
	`(c) Copyright 2015, Cisco Systems.`
	`NPT3000 BIOS v.3.0.2, Tue 05/26/2015`
	
	`Press TAB in 1 seconds to list all boot options`
	`Any other key to active boot...`
	`Press  ctrl L to go to loader prompt in 2 secs`
	
	`Booting kickstart image: bootflash:/n9000-dk9.7.0.3.I2.0.527.bin`
	`Image valid`
	`INIT: version 2.88 booting`
	`Skipping ata_piix for n3k.`
	`Unsquashing rootfs ...`
	
	`Loading IGB driver ...`
	`Installing SSE module ... done`
	`Creating the sse device node ... done`
	`Loading I2C driver ...`
	`Installing CCTRL driver for card_type 31 ...`
	`CCTRL driver for card_index 11081 ...`
	`7.46: Interrupt throttling disabled. No cctrl irq detected.`
	
	`Checking all filesystems./etc/rc.d/rcS.d/S08check-flash-noinit: line 167: sg_inq: command not found`
	`/etc/rc.d/rcS.d/S08check-flash-noinit: line 168: sg_inq: command not found`
	`Current boot disk sda3..`
	`...Skipping LOGFLASH check for N3k...`
	`.Skipping plog check for N3k...`
	
	`Skipping installing default sprom values...`
	`Configuring network ...`
	`Installing LC netdev ...`
	`Installing veobc ...`
		Reloads the switch.

	Command or Action	Purpose
	Installing OBFL driver ... ..done Wed Jan 9 03:59:36 UTC 2002 tune2fs 1.42.1 (17-Feb-2012) Setting reserved blocks percentage to 0% (0 blocks) Starting portmap daemon... creating NFS state directory: done starting 8 nfsd kernel threads: done starting mountd: done starting statd: done Saving image for img-sync ... Loading system software Installing local RPMS Patch Repository Setup completed successfully Uncompressing system image: Wed Jan 9 03:59:46 UTC 2002 blogger: nothing to do. ..done Wed Jan 9 03:59:46 UTC 2002 Creating /dev/mcelog Starting mcelog daemon Removing dme lib Moving N3K specific syslog config file INIT: Entering runlevel: 3 Running S93thirdparty-script... Populating conf files for hybrid sysmgr ... Starting hybrid sysmgr ... 2002 Jan 9 03:59:54 %$ VDC-1 %$ Jan 9 03:59:52 %KERN-2-SYSTEM_MSG: [9.062765] Initializing NVRAM Block 6 - kernel 2002 Jan 9 03:59:54 %$ VDC-1 %$ Jan 9 03:59:52 %KERN-2-SYSTEM_MSG: [10.469175] hwport mode=6type 2. mod_no 0, inst_no 0 - kernel 2002 Jan 9 03:59:58 %$ VDC-1 %$ %USER-0-SYSTEM_MSG: after syslog open - clis 2002 Jan 9 03:59:58 %$ VDC-1 %$ %USER-0-SYSTEM_MSG: after ksink_get_rsw_sched_policy - clis 2002 Jan 9 03:59:58 %$ VDC-1 %$ %USER-0-SYSTEM_MSG: after clis_process_options - clis 2002 Jan 9 03:59:58 %$ VDC-1 %$ %USER-0-SYSTEM_MSG: before access to bkout_cfg - clis 2002 Jan 9 03:59:58 %$ VDC-1 %$ %USER-2-SYSTEM_MSG: main 2348- Done with Shm..Now read commandfiles - clis 2002 Jan 9 03:59:59 %$ VDC-1 %$ %PLATFORM-2-PS_FAIL: Power supply 1 failed or shut down (Serial number N/A) 2002 Jan 9 03:59:59 %$ VDC-1 %$ %PLATFORM-2-PS_OK: Power supply 2 ok (Serial number) 2002 Jan 9 03:59:59 %$ VDC-1 %$ %PLATFORM-2-PS_FANOK: Fan in Power supply 2 ok	

	Command or Action	Purpose
	2002 Jan 9 03:59:59 %$ VDC-1 %$ %PLATFORM-2-PS_ABSENT: Power supply 1 is absent/shutdown, ps-redundancy might be affected 2002 Jan 9 03:59:59 %$ VDC-1 %$ %PLATFORM-2-PS_RED_MODE_CHG: Power supply operational redundancy mode changed to non-redundant 2002 Jan 9 03:59:59 %$ VDC-1 %$ %PLATFORM-2-FANMOD_FAN_OK: Fan module 1 (Fan1(sys_fan1) fan) ok 2002 Jan 9 03:59:59 %$ VDC-1 %$ %PLATFORM-2-FANMOD_FAN_OK: Fan module 2 (Fan2(sys_fan2) fan) ok 2002 Jan 9 03:59:59 %$ VDC-1 %$ %PLATFORM-2-FANMOD_FAN_OK: Fan module 3 (Fan3(sys_fan3) fan) ok 2002 Jan 9 03:59:59 %$ VDC-1 %$ %PLATFORM-2-FANMOD_FAN_OK: Fan module 4 (Fan4(sys_fan4) fan) ok 2002 Jan 9 04:00:01 %$ VDC-1 %$ %USER-2-SYSTEM_MSG: IP Netlink thread init successful - netstack 2002 Jan 9 04:00:08 %$ VDC-1 %$ %USER-2-SYSTEM_MSG: main :2355- Done with reading commandfiles - clis 2002 Jan 9 04:00:18 %$ VDC-1 %$ %USER-0-SYSTEM_MSG: end of default policer - copp 2002 Jan 9 04:00:18 %$ VDC-1 %$ %COPP-2-COPP_NO_POLICY: Control-plane is unprotected. 2002 Jan 9 04:00:27 %$ VDC-1 %$ icmpv6: IPV6 Netlink thread init successful 2002 Jan 9 04:00:28 %$ VDC-1 %$ %VDC_MGR-2-VDC_ONLINE: vdc 1 has come online Waiting for system online status before starting POAP ... 2002 Jan 9 04:01:01 switch %$ VDC-1 %$ %ASCII-CFG-2-CONF_CONTROL: System ready Starting Auto Provisioning ... 2002 Jan 9 04:01:02 switch %$ VDC-1 %$ %USER-0-SYSTEM_MSG: ETH_PORT_UP - port_client Done Abort Auto Provisioning and continue with normal setup ?(yes/no)[n]: 2002 Jan 9 04:01:03 switch %$ VDC-1 %$ %POAP-2-POAP_INITED: POAP process initialized yes ---- System Admin Account Setup ---- Do you want to enforce secure password standard (yes/no) [y]: no Enter the password for "admin": Confirm the password for "admin": ---- Basic System Configuration	

	Command or Action	Purpose
	Dialog VDC: 1 ---- This setup utility will guide you through the basic configuration of the system. Setup configures only enough connectivity for management of the system. Please register Cisco Nexus3000 Family devices promptly with your supplier. Failure to register may affect response times for initial service calls. Nexus3000 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. Would you like to enter the basic configuration dialog (yes/no): no 2015 Jan 9 04:01:26 switch %$ VDC-1 %$ %COPP-2-COPP_POLICY: Control-Plane is protected with policy copp-system-p-policy-strict. User Access Verification switch login: admin Password: Cisco Nexus Operating System (NX-OS) Software TAC support: http://www.cisco.com/tac Copyright (C) 2002-2015, Cisco and/or its affiliates. All rights reserved. The copyrights to certain works contained in this software are owned by other third parties and used and distributed under their own licenses, such as open source. This software is provided "as is," and unless otherwise stated, there is no warranty, express or implied, including but not limited to warranties of merchantability and fitness for a particular purpose. Certain components of this software are licensed under the GNU General Public License (GPL) version 2.0 or GNU General Public License (GPL) version 3.0 or the GNU Lesser General Public License (LGPL) Version 2.1 or Lesser General Public License (LGPL) Version 2.0. A copy of each such license is available at http://www.opensource.org/licenses/gpl-2.0.php and http://opensource.org/licenses/gpl-3.0.html and http://www.opensource.org/licenses/lgpl-2.1.php	

	Command or Action	Purpose
	and http://www.gnu.org/licenses/old-licenses/library.txt.	

Step 5	(Optional) switch(config)# **show system switch-mode**
	Example:
	`switch(config)# show system switch-mode`
	`system switch-mode n9k`
	`switch(config)#`
		Verifies the configuration mode as N9K on the switch.

Managing Users

You can display information about users logged into the device and send messages to those users.

Displaying Information about the User Sessions

You can display information about the user session on the device.

Procedure

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	Displays the user sessions.

Sending a Message to Users

You can send a message to active users currently using the device CLI.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **show users** **Example:** `switch# show users`	Displays the active user sessions.
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.

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

Default Settings for Basic Device Parameters

This table lists the default settings for basic device parameters.

Table 15: Default Basic Device Parameters

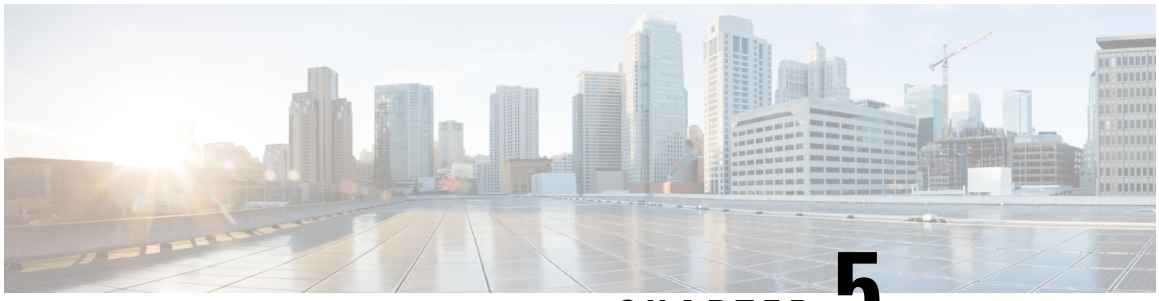
Parameters	Default
MOTD banner text	User Access Verification
Clock time zone	UTC

Additional References for Basic Device Management

You can find additional information related to basic device management.

Related Documents for Basic Device Management

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



CHAPTER 5

Using Power On Auto Provisioning

This chapter contains the following sections:

- Information About PowerOn Auto Provisioning, on page 57
- Guidelines and Limitations for POAP, on page 68
- Setting Up the Network Environment To Use POAP, on page 69
- Configuring a Switch Using POAP, on page 70
- Verifying the Device Configuration, on page 71
- Related Documents for POAP, on page 71

Information About PowerOn Auto Provisioning

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

When a Cisco Nexus Series switch with the POAP feature boots and does not find the startup configuration, the switch enters POAP mode and checks for a USB device containing the configuration script file. If it finds one, it checks that device to see if it also contains the software image files and the switch configuration file.

If the switch does not find a USB device, or if the USB device does not contain the needed image files or switch configuration file, the switch also locates a DHCP server and bootstraps itself with its interface IP address, gateway, and DNS server IP addresses. The switch then obtains the IP address of a TFTP server or the URL of an HTTP server from which it downloads the necessary configuration files.

Note The DHCP information is used only during the POAP process if any configuration files are unavailable on the USB device.

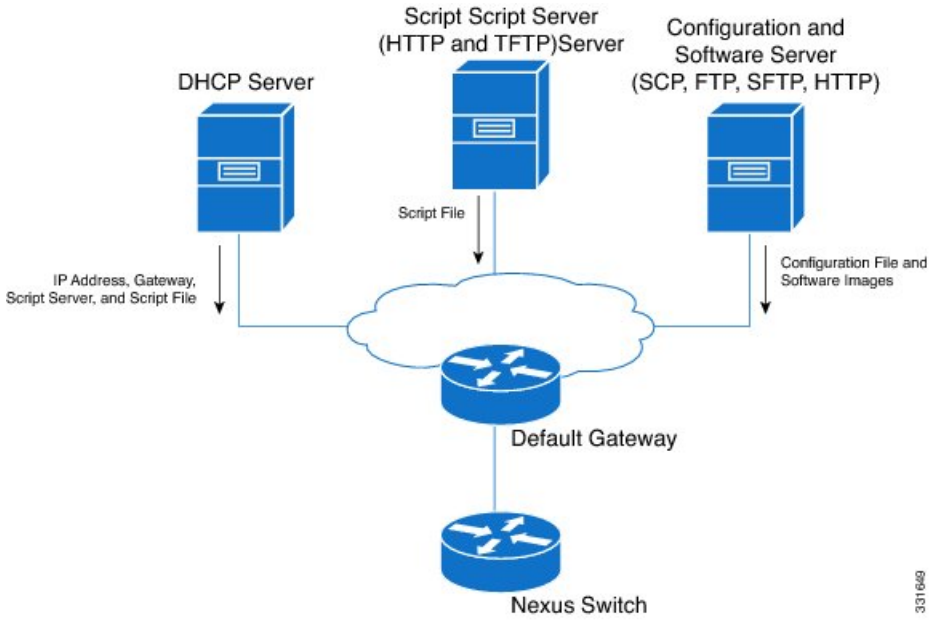
If the backup configuration file does not have the administrative username and the password, POAP causes a console lockout after completion. It is a mandatory step to add the username and the password in the configuration file.

Network Requirements for POAP

If a USB device containing the required installation files is not available, POAP requires the following network infrastructure:

- A DHCP server to bootstrap the interface IP address, gateway address, DNS server, and log server
- A TFTP or HTTP server containing the configuration script used to automate the software image installation and configuration process
- One or more servers containing the desired software images and configuration files

Figure 1: POAP Network Infrastructure



POAP Configuration Script

We provide sample configuration scripts that are developed using the Python programming language. You can customize these scripts to meet the requirements of your network environment.

A complete POAP script can be found at https://github.com/datacenter/nexus9000/blob/master/nx-os/poap/poap.py

For information about customizing this script using Python, see the *Cisco NX-OS Python API Reference Guide* for your platform.

Using the POAP Script and POAP Script Options

Before using the POAP script, perform the following actions:

1. Edit the options dictionary at the top of the script to ensure all the relevant options for your setup are included in the script. Do not change the defaults (in the default options function) directly.
2. Update the MD5 checksum of the POAP script as shown using shell commands.

```
f=poap_nexus_script.py ; cat $f | sed '/^#md5sum/d' > $f.md5 ; sed -i
"s/^#md5sum=.*/#md5sum=\"$(md5sum $f.md5 | sed 's/ .*//')\"/" $f
```

3. If the device has a startup configuration, perform write erase and reload the device.

The following is a comprehensive list of POAP script options and can be specified to alter the POAP script behavior. When downloading from a server, 'hostname', 'username', and 'password' are required. For every mode except 'personality', the 'target_system_image' is also required. Required parameters are enforced by the script and the script will abort if the required parameters are not present. Every option except 'hostname', 'username', and 'password' has a default option. If you do not specify the option in the options dictionary, the default as mentioned here will be used.

- **username**

  The username to use when downloading files from the server. This is not required when using USB.

- **password**

  The password to use when downloading files from the server. This is not required when using USB.

- **hostname**

  The name or address of the server to download files from. This is not required when using USB.

- **mode**

  Default is **serial_number**.

  Use one of the following options:

  1. **personality**

     A method to restore the switch from a tarball.

  2. **serial_number**

     The serial number of the switch to determine the configuration filename. The format for the serial number in the configuration file is conf.*serialnumber*. Example: conf.FOC123456

  3. **hostname**

     The hostname as received in the DHCP options to determine the configuration filename. The format for using the hostname in the configuration file is **conf_<hostname>.cfg** Example: conf_3164-RS.cfg

  4. **mac**

     The interface MAC address to determine the configuration filename. The format for using the hostname in the configuration file is **conf_<macaddress>.cfg** Example: conf_7426CC5C9180.cfg

  5. **raw**

     The configuration filename is used exactly as provided in the options. The filename is not altered in any way.

  6. **location**

     The CDP neighbors are used to determine the configuration filename. The The format for using the location in the configuration file is **conf_<host>_<intf>.cfg** where <host> is the host connected to the device over the POAP interface, and <intf> is the remote interface the POAP interface is connected to. Example: conf_remote-switch_Eth1_8.cfg

- **required_space**

The required space in KB for that particular iteration of POAP. Default is 100000. For multi-step upgrades, specify the size of the last image in the upgrade path of the target image or images.

- **transfer_protocol**

  Any transfer protocol such as http, https, ftp, scp, sftp or tftp that is supported by VSH. Default is scp.

- **config_path**

  The path to the configuration file on the server. Example: /tftpboot. Default is /var/lib/tftpboot.

- **target_system_image**

  The name of the image to download from the remote server. This is the image you get after POAP completes. This is a required parameter for every mode except 'personality'. Default is "".

- **target_image_path**

  The path to the image on the server. Example: /tftpboot. Default is /var/lib/tftpboot.

- **target_kickstart_image**

  The name of the kickstart image to download from the remote server. This is the kickstart image you get after POAP completes. Default is "".

- **destination_path**

  The path to download images and MD5 sums to. Default is /bootflash.

- **destination_system_image**

  The name for the destination system image file name. If not specified, the default will be the 'target_system_image' name.

- **destination_kickstart_image**

  The name for the destination kickstart image file name. If not specified, the default will be the 'target_kickstart_image' name.

- **user_app_path**

  The path on the server where the user scripts, agents, and user data are located. Default is /var/lib/tftpboot.

- **disable_md5**

  This is True if MD5 checking should be disabled. Default is False.

- **midway_system_image**

  The name of the system image to use for the midway system upgrade. By default, the POAP script finds the name of any required midway images in the upgrade path and uses them. Set this option if you prefer to pick a different midway image for a two-step upgrade. Default is "".

- **midway_kickstart_image**

  The name of the kickstart image to use for the midway system upgrade. By default, the POAP script finds the name of any required midway images in the upgrade path and uses them. Set this option if you prefer to pick a different midway image for a two-step upgrade. Default is "".

- **usb_slot**

  The USB slot number to use for USB POAP. Default is 1.

- **source_config_file**

  The name of the configuration file when 'raw' mode is used. Default is poap.cfg

- **vrf**

  The VRF to use for downloads and so on. The VRF is automatically set by the POAP process. Default is the 'POAP_VRF' environment variable.

- **destination_config**

  The name to use for the downloaded configuration. Default is poap_replay.cfg

- **split_config_first**

  The name to use for the first configuration portion if the configuration needs to be split. Applicable only when certain configuration requires a reload to take effect. Default is poap_1.cfg

- **split_config_second**

  The name to use for the second configuration portion if the configuration is split. Default is poap_2.cfg

- **timeout_config**

  The timeout in seconds, for copying the configuration file. Default is 120. For non-legacy images, this option is not used and the POAP process will timeout. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **timeout_copy_system**

  The timeout in seconds, for copying the system image. Default is 2100. For non-legacy images, this option is not used and the POAP process will timeout. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **timeout_copy_kickstart**

  The timeout in seconds, for copying the kickstart image. Default is 900. For non-legacy images, this option is not used and the POAP process will timeout. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **timeout_copy_personality**

  The timeout in seconds, for copying the personality tarball. Default is 900. For non-legacy images, this option is not used and the POAP process will timeout. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **timeout_copy_user**

  The timeout in seconds, for copying any user scripts and agents. Default is 900. For non-legacy images, this option is not used and the POAP process will timeout. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **personality_path**

The remote path to download the personality tarball from. Once the tarball is downloaded and the personality process is started, personality will download all files in the future from locations specified inside the tarball configuration. Default is /var/lib/tftpboot

- **source_tarball**

  The name of the personality tarball to download. Default is personality.tar

- **destination_tarball**

  The name for the downloaded personality tarball after it is downloaded. Default is personality.tar

- **compact_image**

  To copy the compact image set this value to "True" as shown below. The default value is "False".

  ```
  "compact_image" : True
  ```

  For more details on the compact image, see the Compact Image for Cisco Nexus 3000 section in Cisco Nexus 3000 Series NX-OS Software Upgrade and Downgrade Guide.

POAP Upgrade Path

On a Cisco Nexus 3000 Series switch, moving from one software image to another requires intermediate upgrade steps that are handled as part of POAP. You need to know the upgrade path you require to avoid confusions while POAP loads multiple software images.

For the N3K-C3048TP-1GE-SUP platform, if you are using software versions older than Cisco NX-OS Release 5.0(3)U5(1), upgrade to Cisco NX-OS Release 5.0(3)U5(1) first, then upgrade to Cisco NX-OS Release 6.0(2)U6(2a), and finally upgrade to 6.0(2)U6(7) or a latest release.

If you like to skip this extra step of upgrading to Cisco NX-OS Release 6.0(2)U6(2a), you need to use the **midway_system_image** and **midway_kickstart_image** options.

The required software images for each step are:

- Cisco Release 5.0(3)U5(1) - n3000-uk9-kickstart.5.0.3.U5.1.bin, n3000-uk9.5.0.3.U5.1.bin
- Cisco Release 6.0(2)U6(2a) - n3000-uk9-kickstart.6.0.2.U6.2a.bin, n3000-uk9.6.0.2.U6.2a.bin
- Cisco Release 6.0(2)U6(7) - n3000-uk9-kickstart.6.0.2.U6.7.bin, n3000-uk9.6.0.2.U6.7.bin

For all other Cisco Nexus 3xxx platforms, if you are using software versions older than Cisco NX-OS Release 5.0(3)U5(1), upgrade to Cisco NX-OS Release 5.0(3)U5(1) first, then upgrade to 6.0(2)U6(7) or a latest release.

However, if you have a mix of N3K-C3048TP-1GE-SUP platform and other devices, you need to use the default upgrade path (no options needed) for the N3K-C3048TP-1GE-SUP platform, and a different script with the **midway** options set for your other devices. You can use the Vendor Class ID to distinguish between the type of devices.

POAP Upgrade Bootflash Size

For Cisco Nexus 3xxx devices having small bootflashes (< 1.6 GB) requiring multi-step upgrade process may have problems if bootflash is not sufficiently empty. The total amount of bootflash needed can be calculated using the following:

(size of images currently running) + (size of largest midway images) + (size of target images) + (18% of bootflash size)

For example, if you are on a N3K-C3048TP-1GE-SUP platform with 1.6GB of bootflash, running 5.0(3)U5.1.bin, and you are attempting to upgrade to Cisco Release 7.0(3)I4(5), you would require:

5.0(3)U5(1) - current image, 6.0(2)U6(7) - largest midway image, 7.0(3)I4(3) - target image, and 18% of 1.6GB bootflash

(126MB system + 24MB kickstart) + (198MB system + 37MB kickstart) + (665MB NX-OS) + (295MB free space) = 1.31GB

In this example, you have only 200MB on bootflash apart from your current software image.

Setting up the DHCP Server without DNS for POAP

Starting with NX-OS release 7.0(3)I4(5), tftp-server-name can be used without the DNS option. To enable POAP functionality without DNS on releases earlier than 7.0(3)I4(5) release, a custom option (150) must be used to specify the tftp-server-address.

To use the tftp-server-address option, specify the following at the start of your **dhcpd.conf** file.

```
option tftp-server-address code 150 = ip-address;
```

For example:

```
host MyDevice {
    option dhcp-client-identifier "\000SAL12345678";
    fixed-address 2.1.1.10;
    option routers 2.1.1.1;
    option host-name "MyDevice";
    option bootfile-name "poap_nexus_script.py";
    option tftp-server-address 2.1.1.1;
}
```

Downloading and Using User Data, Agents, and Scripts as part of POAP

Under the options dictionary, you can find the **download_scripts_and_agents** function. If you choose to download user scripts and data, uncomment the first **poap_log** line and then use a series of **download_user_app** function calls to download each application. Since older Cisco NX-OS versions do not support recursive copy of directories, such directories must be put into a tarball (TAR archive) and then unpacked once on the switch. The parameters for the **download_scripts_and_agents** function are as follows:

- **source_path** - The path to where the file or tarball is located. This is a required parameter. Example: /var/lib/tftpboot.
- **source_file** - The name of the file to download. This is a required parameter. Example: agents.tar, script.py, and so on.
- **dest_path** - The location to download the file on the switch. Any directories that do not exist earlier will be created. This is an optional parameter. The default is /bootflash.
- **dest_file** - The name to give the downloaded file. This is an optional parameter. The default is unchanged source_file.
- **unpack** - Indicates whether a tarball exists for unpacking. Unpacking is done with **tar -xf** *tarfile* **-C /bootflash**. This is an optional parameter. The default is False.

- **delete_after_unpack** - Indicates whether to delete the downloaded tarball after unpack is successful. There is no effect if unpack is False. The default is False.

Using the download functionality, you can download all the agents and files needed to run POAP. To start the agents, you should have the configuration present in the running configuration downloaded by POAP. Then the agents, scheduler, and cron entry, along with EEM, can be used.

Troubleshooting for POAP

The following is a list of known issues and suggestions while using POAP:

- Issue: POAP script execution fails immediately with no syslogs or output except for a "Script execution failed" statement.

  Suggestion: Use the **python** *script-name* command on the server and make sure there are no syntax errors. The options dictionary is a Python dictionary so each entry must be comma separated and have the key or option and the value separated by a colon.

- Issue: A TypeError exception occurs at various places depending on the incorrectly used option.

  Suggestion: Some options use integers (for example, timeouts and other numeric values). Check the options dictionary for numeric values that are enclosed in quotes. Refer to the options list for the correct usage.

- Issue: POAP over USB is not finding the files that are present.

  Suggestion: Some devices have two USB slots. If you are using USB slot 2, you need to specify that as an option. For more details, see the options under Using the POAP Script and POAP Script Options, on page 58.

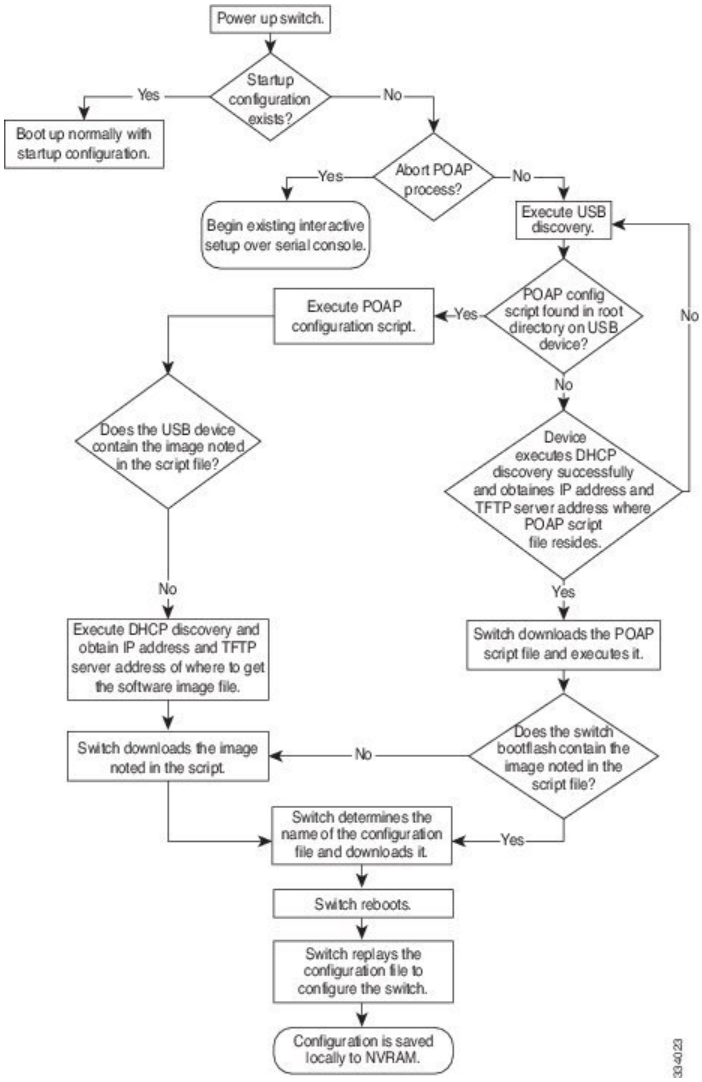
POAP Process

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 2: POAP Process



Power-Up Phase

When you power-up a switch for the first time, it loads the software image installed at manufacturing and tries to find a configuration file to apply after the switch boots. When no configuration file is found, POAP mode starts.

During startup, a prompt appears asking if you want to abort POAP and continue with 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 Layer 2 mode, which ensures that the device does not participate in any Layer 2 forwarding.

USB Discovery Phase

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

If the configuration script file is found on a USB device, POAP begins running the configuration script. If the configuration 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 all of the active interfaces (including the mgmt interface) soliciting DHCP offers from the DHCP server or servers. The DHCP client on the Cisco Nexus switch uses the switch serial number or its MAC address 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 file name, 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 DHCP negotiation.

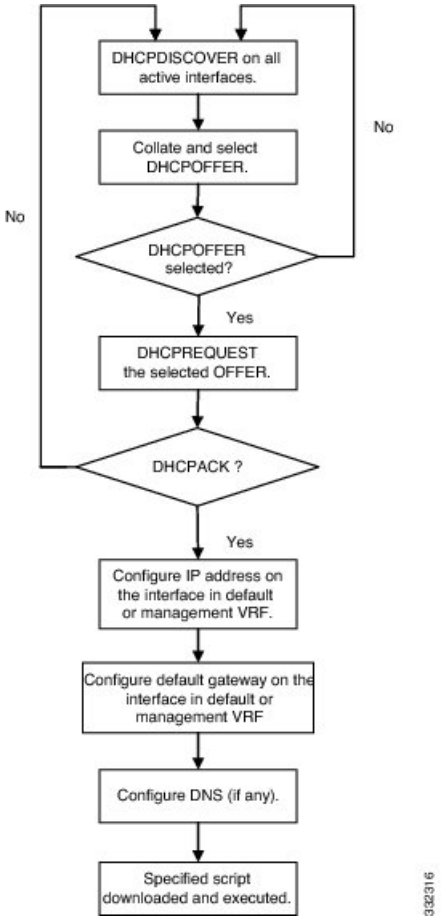
The DHCP discover message also mandates some of the options and solicits these options from the DHCP server after receiving the DHCP OFFER from the DHCP server.

- Option 66 (TFTP server name) or Option 150 (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.
- IP address
- Default gateway
- Option 67 (Bootfile name)—The DHCP server relays the bootfile name to the DHCP client. The bootfile name includes the complete path to the bootfile on the TFTP server. The DHCP client uses this information to download the script file.

When multiple DHCP offers that meet the requirement are received, an offer is randomly chosen. The device completes the DHCP negotiation (request and acknowledgment) with the selected DHCP server, and the DHCP server assigns an IP address to the switch. If there is a failure 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. The POAP process is reinitiated until it succeeds or you manually abort the POAP process.

Figure 3: DHCP Discovery Phase



POAP Dynamic Breakout

Beginning with Cisco NX-OS Release 7.0(3)I4(1), POAP dynamically breaks out ports in an effort to detect a DHCP server behind one of the broken-out ports. Previously, the DHCP server used for POAP had to be directly connected to a normal cable because breakout cables were not supported.

POAP determines which breakout map (for example, 10gx4, 50gx2, 25gx4, or 10gx2) will bring up the link connected to the DHCP server. If breakout is not supported on any of the ports, POAP skips the dynamic breakout process. After the breakout loop completes, POAP proceeds with the DHCP discovery phase as normal.

Note For more information on dynamic breakout, see the interfaces configuration guide for your device.

Script Execution Phase

Once the device has bootstrapped itself using the information in the DHCP acknowledgement, the switch downloads the script file from the TFTP server or the HTTP 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 currently running 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 one 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 clears the existing configuration if any, and applies (replays) the configuration on the upgraded software image. Afterward, the switch copies the running configuration to the startup configuration.

Guidelines and Limitations for POAP

- The Cisco Nexus switch software image must support POAP for this feature to function.
- POAP can be triggered even when the startup-config is present using the **boot poap enable** command.
- If a LACP Layer 3 port-channel is configured on an uplink device connected to the Cisco Nexus device that is being bootstrapped using POAP, the port-channel is not active because all the member links are in a suspended state. Therefore, the Cisco Nexus device that is being bootstrapped using POAP cannot reach the DHCP server or any other infrastructure device needed for POAP. To work around this issue, configure a static L3 port-channel on the uplink device connected to the Cisco Nexus device that is being bootstrapped using POAP.
- If you use POAP to bootstrap a Cisco Nexus device that is a part of a vPC pair using static port-channels on the VPC links, the Cisco Nexus device activates all of its links upon POAP startup. 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 connected to the Cisco Nexus device, and the traffic would be lost.

  To work around this issue, you can configure 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 Series 7000 device through a LACP port-channel, the Cisco Nexus 7000 Series device defaults to suspend its member port if it cannot bundle it as a part of a port-channel. To work around this issue, configure the Cisco Nexus 7000 Series device to not suspend its member ports using the no lacp suspend-individual command from interface configuration mode.
- Important POAP updates are logged in the syslog and are available from the serial console.
- Critical POAP errors are logged to the bootflash. The filename format is *date-time*_poap_*PID*_[init,1,2].log, where *date-time* is in the YYYYMMDD_hhmmss format and *PID* is the process ID.
- Script logs are saved in the bootflash directory. The filename format is *date-time*_poap_*PID*_script.log, where *date-time* is in the YYYYMMDD_hhmmss format and *PID* is the process ID.

- The Scheduler configuration cannot be replayed using POAP. The reason that the Scheduler configuration cannot be replayed is that it is associated with the user (for example "admin") that was logged in when the Scheduler configuration was created. Because the configuration replay using POAP is not associated with any specific user, the scheduler configuration cannot be replayed and fails.

  Instead of configuring the Scheduler, configure the Embedded Event Manager (EEM). An EEM configuration can be downloaded and replayed using POAP.

- You can bypass password and basic POAP configuration by using the **skip** option at the POAP prompt.

  When you use the **skip** option, no password will be configured for the **admin** user. The **copy running-config startup-config** command will be blocked until a valid password is set for the **admin** user.

- The certificates (for example SSL) or configuration that are needed to be applied to the switch should be present in the configuration file.

- The syntax of the poap_script.py file should be validated using any python validation tool before using the file for POAP. Otherwise, if the poap_script.py file is edited and has a syntax error, the POAP process will exit without giving an error.

- 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 3000 switches.

    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.

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 configuration 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. For more information, see the Setting Up the Network Environment To Use POAP, on page 69 section immediately preceeding this section.

Procedure

Step 1 Install the switch in the network.

Step 2 Power on the switch.

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

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

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

The switch boots, and the POAP process begins.

Step 4 (Optional) If you want to bypass password and basic POAP configuration, enter **skip**.

POAP is aborted and password configuration is skipped.

What to do next

Verify the configuration.

Verifying the Device Configuration

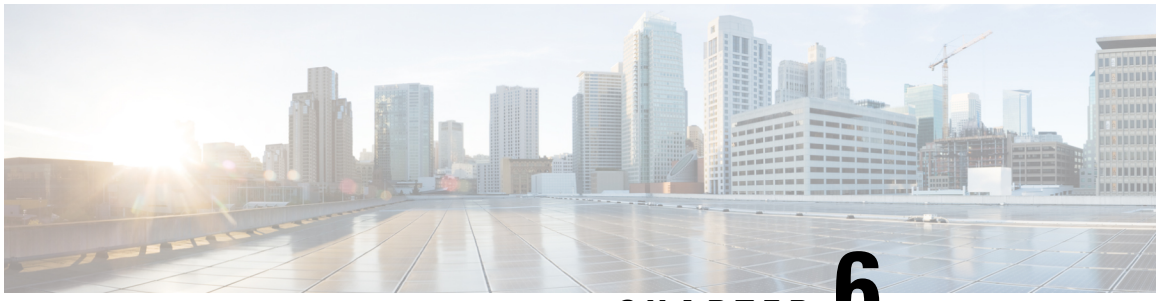
To verify the configuration, use one of the following commands:

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

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

Related Documents for POAP

Related Topic	Document Title
Configuration Script	*Cisco Nexus 3000 Series NX-OS Python API Reference Guide*
DHCP Options and BOOTP Vendor Extensions	RFC2132—http://tools.ietf.org/html/rfc2132
TFTP Server Address Option for DHCPv4	RFC5859—http://tools.ietf.org/html/rfc5859



CHAPTER 6

Using the Device File Systems, Directories, and Files

This chapter contains the following sections:

- Information About the Device File Systems, Directories, and Files, on page 73
- Working with Directories, on page 74
- Working with Files, on page 76
- Working with Archive Files, on page 83
- Examples of Using the File System, on page 86
- Default Settings for File System Parameters, on page 89
- Additional References for File Systems, on page 90

Information 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 16: 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.

File System Name	Module	Description
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 (slot0:, usb1:, and usb2:). You can navigate through these directories and use them for files.

Files

You create and access files on bootflash:. volatile:, slot0:, usb1:, and usb2: file systems. You can only access files on the system: file systems. You can use the debug: file system for debug log files specified in the **debug logfile** command.

You can download files, such as system image files, 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.

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.

Procedure

	Command or Action	Purpose
Step 1	**pwd** **Example:** `switch# pwd`	Displays the name of your current directory.

Changing the Current Directory

You can change the current directory for file system operations. The initial default directory is bootflash:.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.
Step 2	**cd** {*directory* | *filesystem*:[*//module/*][*directory*]} **Example:** `switch# cd slot0:`	Changes to a new current directory. The file system, module, and directory names are case sensitive.

Creating a Directory

You can create directories in the bootflash: and flash device file systems.

Procedure

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

Displaying Directory Contents

You can display the contents of a directory.

Procedure

	Command or Action	Purpose
Step 1	**dir** [*directory* | *filesystem*:[//*module*/][*directory*]] **Example:** `switch# dir bootflash:test`	Displays the directory contents. The default is the current working directory. The file system and directory names are case sensitive.

Deleting a Directory

You can remove directories from the file systems on your device.

Before you begin

Ensure that the directory is empty before you try to delete it.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.
Step 2	(Optional) **dir** [*filesystem* :[//*module*/][*directory*]] **Example:** `switch# dir bootflash:test`	Displays the contents of the current directory. The file system, module, and directory names are case sensitive. If the directory is not empty, you must delete all the files before you can delete the directory.
Step 3	**rmdir** [*filesystem* :[//*module*/]]*directory* **Example:** `switch# rmdir test`	Deletes a directory. The file system and directory name are case sensitive.

Accessing 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.

Procedure

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

Copying Files

You can make copies of files, either within the same directory or on another directory.

While copying a file to an HTTP server, you can use any valid character, such as ~, in the directory or filename. You can also access public_html directories.

Cisco NX-OS Release 6.0(2)U4(1) supports the configuration of a non-default port when copying a file to an HTTP server.

Note Use the **dir** command to ensure that enough space is available in the target file system. If enough space is not available, use the **delete** command to remove unneeded files.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.
Step 2	(Optional) **dir** [*filesystem*:[//*module*/][*directory*]] **Example:** `switch# dir bootflash`	Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 3	**copy** [*filesystem*:[//*module*/][*directory*/] | *directory*/]*source-filename* | {*filesystem*:[//*module*/][*directory*/]] | *directory*/}[*target-filename*] **Example:** `switch# copy test old_tests/test1`	Copies a file. The file system, module, and directory names are case sensitive. The *source-filename* argument is alphanumeric, case sensitive, and has a maximum of 64 characters. If the *target-filename* argument is not specified, the filename defaults to the *source-filename* argument value. The directory names and file names can include any valid character such as ~ while copying a file to an HTTP server. You can also configure a non-default port when copying a file to an HTTP server by specifying the port number with the server, for example **copy http:**[//*server*:*port*] [/*path*] /*filename*. The copy command supports ftp, scp, sftp, tftp and http protocols.

Example

This example shows how to copy a file to an HTTP server using characters such as ~ in the path:

```
switch# copy http://2002:48a3:ca47::48a3:ca47/~users/pat3/patch.45.bin bootflash:t.s vrf
management
>   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
>                                  Dload  Upload   Total   Spent    Left  Speed
> 100  177M  100  177M    0     0  9665k      0  0:00:18  0:00:18 --:--:-- 9600k
> Copy complete, now saving to disk (please wait)...
> switch#
```

Configuring the Source Interface for Copying Files to or from a Remote Server

You can configure a source-interface while copying files to or from a remote server. The source interface can be:

- Ethernet
- Loopback

- Management
- Port Channel
- VLAN

Procedure

	Command or Action	Purpose
Step 1	**copy** *scheme://server/[url/]filename* **source-interface** *type slot/port* **Example:** `copy http://123.45.67.890//index.html bootflash: source-interface ethernet 1/5`	Configures the source interface to be used while copying a file to or from a remote server. For the scheme argument, you can enter **tftp**, **ftp**, **scp**, **http**, or **sftp**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case sensitive.

Copying Files to the HTTP Server

You can copy a file from boot flash, running configuration, or startup configuration to the HTTP server in the default path of server using the new feature HTTP PUT. The HTTP PUT functionality copies a file into the HTTP server via the default or the management VRF. It uses the HTTP POST method to upload the files and HTTP upload can be done via v4/v6 interface to v4/v6 HTTP server.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.
Step 2	Required: **copy bootflash :**///<*filename*> **http:**// <*httpserver-ip*> /*path source-interface* <*interface*> **Example:** `switch# copy bootflash:///<filename> http://httpserver-ip/path source-interface <int>`	Copies the file from the boot flash into the HTTP server via the source interface.
Step 3	Required: **copy bootflash :**///<*filename*> **http:**// <*httpserver-ip*>/*path vrf* <*default/management*> **Example:**	Copies the file from the boot flash into the HTTP server via the default or the management VRF interface.

	Command or Action	Purpose
	`switch# copy bootflash:///<filename> http://httpserver-ip/path vrf <default/management>`	
Step 4	Required: **copy running-config http://***<httpserver-ip>/path vrf <default/management>* **Example:** `switch# copy running-config http://httpserver-ip/path vrf <default/management>`	Copies the running configuration file into the HTTP server via the default or the management VRF interface.
Step 5	Required: **copy startup-config http://***<httpserver-ip>/path vrf <default/management>* **Example:** `switch# copy startup-config http://httpserver-ip/path vrf <default/management>`	Copies the startup configuration file into the HTTP server via the default or the management VRF interface.

Example

This example shows how to copy a file to an HTTP server via the default VRF interface:

```
switch# copy n3000-uk9-kickstart.6.0.2.U5.0.995.bin http://12.1.2.10/httproot vrf default
Enter username: test
Enter host password for user 'test':
301 - Moved permanently to <a href="/httproot/">/httproot/</a>Copy
complete, now saving to disk (please wait)...
switch#
```

Note The script that performs file uploads in the HTTP server should read the file into the 'filename' argument. For example, if it is a php script, use $_FILES['filename'].

Deleting Files

You can delete a file from a directory.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **dir** [*filesystem***:**[*//module/*][*directory*]] **Example:** `switch# dir bootflash`	Displays the contents of the current directory. The file system and directory name are case sensitive.

	Command or Action	Purpose
Step 2	**delete** {*filesystem*:[//*module*/][*directory*/] | *directory*/}*filename* **Example:** `switch# delete test old_tests/test1`	Deletes a file. The file system, module, and directory names are case sensitive. The *source-filename* argument is case sensitive. **Caution** If you specify a directory, the **delete** command deletes the entire directory and all its contents.

Displaying File Contents

You can display the contents of a file.

Procedure

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

Displaying File Checksums

You can display checksums to check the file integrity.

Procedure

	Command or Action	Purpose
Step 1	**show file** [*filesystem*:[//*module*/]][*directory*/]*filename* {**cksum** | **md5sum**} **Example:** `switch# show file bootflash:trunks2.cfg cksum`	Displays the checksum or MD5 checksum of the file.

Compressing and Uncompressing Files

You can compress and uncompress files on your Cisco NX-OS device using Lempel-Ziv 1977 (LZ77) coding.

Procedure

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

Displaying the Last Lines in a File

You can display the last lines of a file.

Procedure

	Command or Action	Purpose
Step 1	**tail** [*filesystem*:[//*module*/]][*directory*/]*filename* [*lines*] **Example:** `switch# tail ospf-gr.conf`	Displays the last lines of a file. The default number of lines is 10. The range is from 0 to 80 lines.

Redirecting show Command Output to a File

You can redirect **show** command output to a file on bootflash:, slot0:, volatile:, or on a remote server.

Procedure

	Command or Action	Purpose
Step 1	*show-command* > [*filesystem*:[//*module*/][*directory*] | [directory /]]*filename* **Example:** `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.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.
Step 2	(Optional) **cd** {*filesystem*:[//*module*/][*directory*] | *directory*} **Example:** `switch# cd bootflash:test_scripts`	Changes the default directory.
Step 3	**find** *filename-prefix* **Example:** `switch# find bgp_script`	Finds all filenames in the default directory and in its subdirectories beginning with the filename prefix. The filename prefix is case sensitive.

Working with Archive Files

The Cisco NX-OS software supports archive files. 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 Files

You can create an archive file and add files to it. You can specify the following compression types:

- bzip2
- gzip
- Uncompressed

The default is gzip.

Procedure

	Command or Action	Purpose
Step 1	**tar create** {**bootflash:** | **volatile:**}*archive-filename* [**absolute**] [**bz2-compress**] [**gz-compress**] [**remove**] [**uncompressed**] [**verbose**] *filename-list*	Creates an archive file and adds files to it. The filename is alphanumeric, not case sensitive, and has a maximum length of 240 characters. The **absolute** keyword specifies that the leading backslash characters (\) should not be removed from the names of the files added to the archive

	Command or Action	Purpose
		file. By default, the leading backslash characters are removed. The **bz2-compress**, **gz-compress**, and **uncompressed** keywords determine the compression utility used when files are added, or later appended, to the archive and the decompression utility to use when extracting the files. If you do not specify an extension for the archive file, the defaults are as follows: • For **bz2-compress**, the extension is .tar.bz2. • For **gz-compress**, the extension is .tar.gz. • For **uncompressed**, the extension is .tar. The **remove** keyword specifies that the Cisco NX-OS software should delete the files from the file system after adding them to the archive. By default, the files are not deleted. The **verbose** keyword specifies that the Cisco NX-OS software should list the files as they are added to the archive. By default, the files are listed as they are added.

Example

This example shows how to create a gzip compressed archive file:

```
switch# tar create bootflash:config-archive gz-compress bootflash:config-file
```

Appending Files to an Archive File

You can append files to an existing archive file on your Cisco NX-OS device.

Before you begin

You have created an archive file on your Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**tar append** {**bootflash:** | **volatile:**}*archive-filename* [**absolute**] [**remove**] [**verbose**] *filename-list*	Adds files to an existing archive file. The archive filename is not case sensitive. The **absolute** keyword specifies that the leading backslash characters (\) should not be removed

	Command or Action	Purpose
		from the names of the files added to the archive file. By default, the leading backslash characters are removed. The **remove** keyword specifies that the Cisco NX-OS software should delete the files from the filesystem after adding them to the archive. By default, the files are not deleted. The **verbose** keyword specifies that the Cisco NX-OS software should list the files as they are added to the archive. By default, the files are listed as they are added.

Example

This example shows how to append a file to an existing archive file:

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

Extracting Files from an Archive File

You can extract files to an existing archive file on your Cisco NX-OS device.

Before you begin

You have created an archive file on your Cisco NX-OS device.

Procedure

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

Example

This example shows how to extract files from an existing archive file:

```
switch# tar extract bootflash:config-archive.tar.gz
```

Displaying the Filenames in an Archive File

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 example of using 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
   12198912     Aug 27 16:29:18 2003  m9500-sf1ek9-kickstart-mzg.1.3.0.39a.bin
    1864931     Apr 29 12:41:59 2003  dplug2
      12288     Apr 18 20:23:11 2003  lost+found/
   12097024     Nov 21 16:34:18 2003  m9500-sf1ek9-kickstart-mz.1.3.1.1.bin
   41574014     Nov 21 16:34:47 2003  m9500-sf1ek9-mz.1.3.1.1.bin

Usage for bootflash://sup-remote
   67747169 bytes used
  116812447 bytes free
  184559616 bytes total
```

This example shows how to delete a file on 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 slot0:samplefile slot0:mystorage/samplefile
```

This example shows how to move a file in the default file system:

```
switch# move samplefile mystorage/samplefile
```

Copying Files

This example shows how to copy the file called samplefile from the root directory of the slot0: file system to the mystorage directory:

```
switch# copy slot0:samplefile slot0:mystorage/samplefile
```

This example shows how to copy a file from the current directory 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:system_image bootflash://sup-2/system_image
```

You can also use the **copy** command to upload and download files from the slot0: or 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.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.
Step 2	(Optional) **dir** [*filesystem* **:**[*//module/*][*directory*]] **Example:** `switch# dir bootflash:test`	Displays the contents of the current directory. The file system, module, and directory names are case sensitive. If the directory is not empty, you must delete all the files before you can delete the directory.
Step 3	**rmdir** [*filesystem* **:**[*//module/*]]*directory* **Example:** `switch# rmdir test`	Deletes a directory. The file system and directory name are case sensitive.

Displaying File Contents

This example shows how to display the contents of a file on an external flash device:

```
switch# show file slot0:test
configure terminal
interface ethernet 1/1
no shutdown
end
show interface ethernet 1/1
```

This example shows how to display the contents of a file that resides in the current directory:

```
switch# show file myfile
```

Displaying File Checksums

This example shows how to display the checksum of a file:

```
switch# show file bootflash:trunks2.cfg cksum
583547619
```

This example shows how to display the MD5 checksum of a file:

```
switch# show file bootflash:trunks2.cfg md5sum
3b94707198aabefcf46459de10c9281c
```

Compressing and Uncompressing Files

This example shows how to compress a file:

```
switch# dir
    1525859     Jul 04 00:51:03 2003 Samplefile
...
switch# gzip volatile:Samplefile
switch# dir
     266069     Jul 04 00:51:03 2003 Samplefile.gz
...
```

This example shows how to uncompress a compressed file:

```
switch# dir
     266069     Jul 04 00:51:03 2003 Samplefile.gz
...
switch# gunzip samplefile
switch# dir
    1525859     Jul 04 00:51:03 2003 Samplefile
...
```

Redirecting show Command Output

This example shows how to direct the output to a file on the bootflash: file system:

```
switch# show interface > bootflash:switch1-intf.cfg
```

This example shows how to direct the output to a file on external flash memory:

```
switch# show interface > slot0:switch-intf.cfg
```

This example shows how to direct the output to a file on a TFTP server:

```
switch# show interface > tftp://10.10.1.1/home/configs/switch-intf.cfg
Preparing to copy...done
```

This example shows how to direct the output of the **show tech-support** command to a file:

```
switch# show tech-support > Samplefile
Building Configuration ...
switch# dir
    1525859     Jul 04 00:51:03 2003 Samplefile
Usage for volatile://
    1527808 bytes used
   19443712 bytes free
   20971520 bytes total
```

Finding Files

This example shows how to find a file in the current default directory:

```
switch# find smm_shm.cfg
/usr/bin/find: ./lost+found: Permission denied
./smm_shm.cfg
./newer-fs/isan/etc/routing-sw/smm_shm.cfg
./newer-fs/isan/etc/smm_shm.cfg
```

Default Settings for File System Parameters

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

Table 17: Default File System Settings

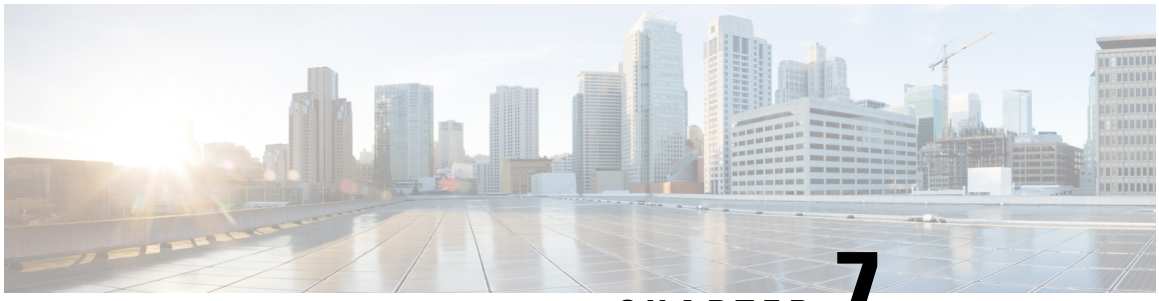
Parameters	Default
Default filesystem	bootflash:

Additional References for File Systems

This section includes additional information related to the file systems.

Related Documents for File Systems

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



CHAPTER 7

Working with Configuration Files

This chapter contains the following sections:

- Information About Configuration Files, on page 91
- Guidelines and Limitations for Configuration Files, on page 92
- Managing Configuration Files, on page 92
- Configuration Archive and Configuration Log, on page 103
- Verifying the Device Configuration, on page 107
- Examples of Working with Configuration Files, on page 107
- Additional References for Configuration Files, on page 108

Information About Configuration Files

Configuration files contain the Cisco NX-OS software commands used to configure the features on a Cisco NX-OS device. Commands are parsed (translated and executed) by the Cisco NX-OS software when the system is booted (from the startup-config file) or when you enter commands at the CLI in a configuration mode.

To change the startup configuration file, you can either save the running-configuration file to the startup configuration using the **copy running-config startup-config** command or copy a configuration file from a file server to the startup configuration.

Types of Configuration Files

The Cisco NX-OS software has two types of configuration files, running configuration and startup configuration. The device uses the startup configuration (startup-config) during device startup to configure the software features. The running configuration (running-config) contains the current changes that you make to the startup-configuration file. The two configuration files can be different. You might want to change the device configuration for a short time period rather than permanently. In this case, you would change the running configuration by using commands in global configuration mode but not save the changes to the startup configuration.

To change the running configuration, use the **configure terminal** command to enter global configuration mode. As you use the Cisco NX-OS configuration modes, commands generally are executed immediately and are saved to the running configuration file either immediately after you enter them or when you exit a configuration mode.

To change the startup-configuration file, you can either save the running configuration file to the startup configuration or download a configuration file from a file server to the startup configuration.

Related Topics

Saving the Running Configuration to the Startup Configuration, on page 93
Downloading the Startup Configuration From a Remote Server, on page 95

Guidelines and Limitations for Configuration Files

Configuration file guidelines and limitations are as follows:

- Beginning with NX-OS 7.0(3)I7(4), the **reload timer** command is supported to enable a reboot after a delay of 5 -60 seconds.

Managing Configuration Files

This section describes how to manage configuration files.

Copying Configuration Files to the Startup Configuration

You can directly copy configuration files, through FTP or SCP, to the startup configuration without reloading the device.

Procedure

	Command or Action	Purpose
Step 1	**copy** *scheme://[user@]server/[url/]filename* **startup-config** **Example:** `switch(boot)# copy scp://user@123.40.56.78/wp/user/abc_cfg startup-config`	Copies the configuration file directly through SCP or FTP to the startup configuration. For the *scheme* argument, you can enter either **ftp** or **scp**. The *user@* argument is your username, the *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *user@*, *server*, *url*, and *filename* arguments are case sensitive. This process does not require you to reload the device.

Configuring the Source Interface for Copying Configuration Files to or from a Remote Server

You can configure a source-interface while copying configuration files to or from a remote server. The source interface can be:

- Ethernet
- Loopback

- Management
- Port Channel
- VLAN

Procedure

	Command or Action	Purpose
Step 1	**copy** *scheme://server/[url/]filename* **source-interface** *type source/port* **Example:** `copy sftp://user@12.345.678.9//wp/user/abc_config . source-interface ethernet 1/5`	Configures the source interface to be used while copying a configuration file to or from a remote server. For the scheme argument, you can enter **tftp**, **ftp**, **scp**, **http**, or **sftp**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case sensitive.

Saving the Running Configuration to the Startup Configuration

You can save the running configuration to the startup configuration to save your changes for the next time you that reload the device.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **show running-config** **Example:** `switch# show running-config`	Displays the running configuration.
Step 2	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Copying a Configuration File to a Remote Server

You can copy a configuration file stored in the internal memory to a remote server as a backup or to use for configuring other Cisco NX-OS devices.

Procedure

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

Example

This example shows how to copy the configuration file to a remote server:

```
switch# copy running-config
tftp://10.10.1.1/sw1-run-config.bak
switch# copy startup-config
tftp://10.10.1.1/sw1-start-config.bak
```

Downloading the Running Configuration From a Remote Server

You can configure your Cisco NX-OS device by using configuration files that you created on another Cisco NX-OS device and uploaded to a remote server. You then download the file from the remote server to your device using TFTP, FTP, Secure Copy (SCP), Secure Shell FTP (SFTP), HTTPS, or HTTP to the running configuration.

Before you begin

Ensure that the configuration file that you want to download is in the correct directory on the remote server.

Ensure that the permissions on the file are set correctly. Permissions on the file should be set to world-read.

Ensure that your Cisco NX-OS device has a route to the remote server. The Cisco NX-OS device and the remote server must be in the same subnetwork if you do not have a router or a default gateway to route traffic between subnets.

Check connectivity to the remote server using the **ping** or **ping6** command.

Procedure

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

Downloading the Startup Configuration From a Remote Server

You can configure your Cisco NX-OS device by using configuration files that you created on another Cisco NX-OS device and uploaded to a remote server. You then download the file from the remote server to your device using TFTP, FTP, Secure Copy (SCP), Secure Shell FTP (SFTP), HTTP, or HTTPS to the startup configuration.

Caution This procedure disrupts all traffic on the Cisco NX-OS device.

Before you begin

Log in to a session on the console port.

Ensure that the configuration file that you want to download is in the correct directory on the remote server.

Ensure that the permissions on the file are set correctly. Permissions on the file should be set to world-read.

Ensure that your Cisco NX-OS device has a route to the remote server. The Cisco NX-OS device and the remote server must be in the same subnetwork if you do not have a router or a default gateway to route traffic between subnets.

Check connectivity to the remote server using the **ping** or **ping6** command.

Procedure

	Command or Action	Purpose
Step 1	**write erase** **Example:** `switch# write erase`	Erases the startup configuration file.

Step 2	**reload**
	Example:
	`switch# reload`
	`This command will reboot the system.`
	`(y/n)?  [n] y`
	`...`
	`Enter the password for "admin":`
	`<password>`
	`Confirm the password for "admin":`
	`<password>`
	`...`
	`Would you like to enter the basic configuration`
	`dialog (yes/no): n`
	`switch#`
		Reloads the Cisco NX-OS device.
		Note Do not use the setup utility to configure the device.

Step 3	**copy** *scheme***://***server*/[*url* /]*filename* **running-config** **Example:** `switch# copy tftp://10.10.1.1/my-config running-config`	Downloads the running configuration file from a remote server. For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, **sftp:**, **http:**, or **https:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case sensitive.
Step 4	**copy running-config startup-config** **Example:**	Saves the running configuration file to the startup configuration file.

	Command or Action	Purpose
	`switch# copy running-config startup-config`	**Note** You can use the **copy** {**ftp:** | **scp:** | **sftp:** | **tftp:** | **http:** | **https:**}*source-url* **startup-config** command to copy a configuration file from a network server to the switch startup configuration. This command replaces the startup configuration file with the copied configuration file. Beginning with Cisco NX-OS Release 6.0(2)U2(1), the startup configuration file is stored as an ASCII text file and all commands in the configuration file are run during the next bootup to generate the binary configuration file. This is equivalent to booting with **write erase** and applying configuration commands individually on reload. Because all commands in the startup configuration file are run as configuration commands, this can delay the ASCII configuration file from taking effect.
Step 5	(Optional) **show startup-config** **Example:** `switch# show startup-config`	Displays the running configuration.

Related Topics

Copying Files, on page 87

Copying Configuration Files to an External Flash Memory Device

You can copy configuration files to an external flash memory device as a backup for later use.

Before you begin

Insert the external Flash memory device into the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **dir** {**slot0:** | **usb1:** | **usb2:**}[*directory/*] **Example:**	Displays the files on the external flash memory device.

	Command or Action	Purpose
	`switch# dir slot0:`	
Step 2	**copy running-config** {**slot0:** | **usb1:** | **usb2:**}[*directory*/]*filename* **Example:** `switch# copy running-config slot0:dsn-running-config.cfg`	Copies the running configuration to an external flash memory device. The *filename* argument is case sensitive.
Step 3	**copy startup-config** {**slot0:** | **usb1:** | **usb2:**}[*directory*/]*filename* **Example:** `switch# copy startup-config slot0:dsn-startup-config.cfg`	Copies the startup configuration to an external flash memory device. The *filename* argument is case sensitive.

Related Topics

Copying Files, on page 87

Copying the Running Configuration from an External Flash Memory Device

You can configure your Cisco NX-OS device by copying configuration files created on another Cisco NX-OS device and saved to an external flash memory device.

Before you begin

Insert the external flash memory device into the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **dir** {**slot0:** | **usb1:** | **usb2:**}[*directory*/] **Example:** `switch# dir slot0:`	Displays the files on the external flash memory device.
Step 2	**copy** {**slot0:** | **usb1:** | **usb2:**}[*directory*/]*filename* **running-config** **Example:** `switch# copy slot0:dsn-config.cfg running-config`	Copies the running configuration from an external flash memory device. The *filename* argument is case sensitive.
Step 3	(Optional) **show running-config** **Example:** `switch# show running-config`	Displays the running configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

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

Related Topics

Copying Files, on page 87

Copying the Startup Configuration from an External Flash Memory Device

You can recover the startup configuration on your Cisco NX-OS device by downloading a new startup configuration file saved on an external flash memory device.

Before you begin

Insert the external flash memory device into the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **dir** {**slot0:** | **usb1:** | **usb2:**}[*directory/*] **Example:** `switch# dir slot0:`	Displays the files on the external flash memory device.
Step 2	**copy** {**slot0:** | **usb1:** | **usb2:**}[*directory/*]*filename* **startup-config** **Example:** `switch# copy slot0:dsn-config.cfg startup-config`	Copies the startup configuration from an external flash memory device. The *filename* argument is case sensitive.
Step 3	(Optional) **show startup-config** **Example:** `switch# show startup-config`	Displays the startup configuration.

Related Topics

Copying Files, on page 87

Copying Configuration Files to an Internal File System

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

Procedure

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

Related Topics

Copying Files, on page 77

Rolling Back to a Previous Configuration

Problems, such as memory corruption, can occur that make it necessary for you to recover your configuration from a backed up version.

Note Each time that you enter a **copy running-config startup-config** command, a binary file is created and the ASCII file is updated. A valid binary configuration file reduces the overall boot time significantly. A binary file cannot be uploaded, but its contents can be used to overwrite the existing startup configuration. The **write erase** command clears the binary file.

Procedure

	Command or Action	Purpose
Step 1	**write erase** **Example:** `switch# write erase`	Clears the current configuration of the switch.
Step 2	**reload** **Example:** `switch# reload`	Restarts the device. You will be prompted to provide a kickstart and system image file for the device to boot and run.

	Command or Action	Purpose
		Note By default, the **reload** command reloads the device from a binary version of the startup configuration. Beginning with Cisco NX-OS 6.2(2), you can use the **reload ascii** command to copy an ASCII version of the configuration to the start up configuration when reloading the device.
Step 3	**copy** *configuration_file* **running-configuration** **Example:** `switch# copy bootflash:start-config.bak running-configuration`	Copies a previously saved configuration file to the running configuration. **Note** The *configuration_file* filename argument is case sensitive.
Step 4	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the start-up configuration.

Erasing a Configuration

You can erase the configuration on your device to return to the configuration defaults. "Configuration" refers to the startup configuration as seen in 'show startup'. No other internal application or process states are cleared.

You can erase the following configuration files saved in the persistent memory on the device:

- Startup
- Boot
- Debug

The **write erase** command erases the entire startup configuration, except for the following:

- Boot variable definitions
- The IPv4 configuration on the mgmt0 interface, including the following:
  - Address
  - Subnet mask

To remove the boot variable definitions and the IPv4 configuration on the mgmt0 interface, use the **write erase boot** command. To remove all application persistency files such as patch rpms, third party rpms, application configuration in /etc directory other than configuration, use 'install reset'.

Important POAP is enabled by default, and it will prevent you from using the switch after you use the **write erase** and **reload** commands. To use the switch after erasing the configuration on the switch and reloading it, ensure that you have access to the console.

Procedure

	Command or Action	Purpose

Step 1	**write erase** [**boot** | **debug**]
	Example:
	`switch# write erase`
	`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 Quality of Service (QoS) and/or access control list (ACL) configurations.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **show running-config** *type* **inactive-if-config** **Example:** `# show running-config ipqos inactive-if-config`	Displays any inactive ACL or QoS configurations. The values for the *type* argument are **aclmgr** and **ipqos**. • **aclmgr**— Displays any inactive configurations for aclmgr. • **ipqos**—Displays any inactive configurations for qosmgr.

Step 2	**clear inactive-config** *policy*
	Example:
	`# clear inactive-config qos`
	`clear qos inactive config`
	`Inactive if config for QoS manager is saved at/bootflash/qos_inactive_if_config.cfg`
	`for vdc default & for other than default`
		Clears inactive configurations.
		The values for the *policy* argument are **qos** and **acl**.
		The following describes the values:
		• **qos**—Clears inactive QoS configurations.
		• **acl**— Clears inactive ACL configurations.

	Command or Action	Purpose
	vdc: /bootflash/vdc_x/qos_inactive_if_config.cfg (where x is vdc number) you can see the log file @ show inactive-if-config log	• **acl qos**—Clears inactive ACL configurations and inactive QoS configurations.
Step 3	(Optional) **show inactive-if-config log** **Example:** # show inactive-if-config log	Displays the commands that were used to clear the inactive configurations.

Configuration Archive and Configuration Log

This section contains information on configuration archive and configuration log.

Information About Configuration Archive

The configuration archive is intended to provide a mechanism to store, organize, and manage an archive of the configuration files to enhance the configuration rollback capability provided by the **configure replace** command. Before configuration archiving was introduced, you could save copies of the running configuration using the **copy running-config** *destination-url* command, storing the replacement file either locally or remotely. However, this method lacked any automated file management. The configuration replace and configuration rollback provides the capability to automatically save copies of the running configuration to the configuration archive. These archived files serve as checkpoint configuration references and can be used by the **configure replace** command to revert to the previous configuration states.

The **archive config** command allows you to save configurations in the configuration archive using a standard location and filename prefix that is automatically appended with an incremental version number (and optional timestamp) as each consecutive file is saved. This functionality provides a means for consistent identification of saved configuration files. You can specify how many versions of the running configuration are kept in the archive. After the maximum number of files are saved in the archive, the oldest file is automatically deleted when the next, most recent file is saved. The **show archive** command displays information for all configuration files saved in the configuration archive.

The configuration archive, wherein the configuration files are stored and are available for use with the **configure replace** command, can be located on the following file systems:

- If your platform has disk0--disk0:, disk1:, ftp:, pram:, rcp:, slavedisk0:, slavedisk1:, or tftp:
- If your platform does not have disk0--bootflash:, tftp:,, and ftp:

Note The TFTP and FTP for this feature use VRF management.

Configuring the Characteristics of the Configuration Archive

Before using the **archive config** command, the configuration archive must be configured. Complete the following steps to configure the characteristics of the configuration archive:

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal** **Example:** `switch# configure terminal`	Enters the global configuration mode.
Step 2	switch(config)# **archive** **Example:** `switch(config)# archive`	Enters the archive configuration mode.
Step 3	switch(config-archive)# **path** *url* **Example:** `switch(config-archive)# path bootflash:myconfig`	Specifies the location and the filename prefix for the files in the configuration archive. • Depending on your hardware platform, the name of your file system can be different than the one displayed in the example. **Note** If a directory is specified in the path instead of the file, the directory name must be followed by a forward slash as follows: path flash:/*directory*/. The forward slash is not necessary after a filename; it is necessary only when specifying a directory.
Step 4	switch(config-archive)# **maximum** *number* **Example:** `switch(config-archive)# maximum 14`	(Optional) Sets the maximum number of archive files of the running configuration to be saved in the configuration archive. • The *number* is the maximum number of the archive files of the running configuration that can be saved in the configuration archive. The range is 1 to 14. The default is 10. **Note** Before using this command, you must configure the **path** to specify the location and filename prefix for the files in the configuration archive.
Step 5	switch(config-archive)# **time-period** *minutes* **Example:** `switch(config-archive)# time-period 10`	(Optional) Sets the time increment for automatically saving an archive file of the current running configuration in the configuration archive. • The *minutes* argument specifies how often, in minutes, to automatically save an

	Command or Action	Purpose
		archive file of the current running configuration in the configuration archive. **Note** Before using this command, you must configure the **path** command to specify the location and filename prefix for the files in the configuration archive.
Step 6	switch# **archive config** **Example:** `switch# archive config`	Saves the current running configuration file to the configuration archive. **Note** You must configure the **path** before using the **archive config** command.
Step 7	switch#**show archive log config all**	Displays the configuration log entries for all the users.

Information About Configuration Log

The configuration change notification and logging tracks the changes that are made to the running configuration by maintaining a configuration log. This configuration log tracks the changes that are initiated only through the CLI or HTTP. Only complete commands that result in the invocation of action routines are logged. The following types of entries are not logged:

- Commands that result in a syntax error message
- Partial commands that invoke the device help system

The configuration log tracks the changes that are initiated only through the CLI. For each configuration command that is executed, the following information is logged:

- A configuration change sequence number
- The line from which the command was executed
- The name of the user that executed the command
- The command that was executed

You can display the information from the configuration log by using the **show archive log config all** command

For each configuration command that is executed, the following information is logged:

- The command that was executed
- The configuration mode in which the command was executed
- The name of the user that executed the command
- The time at which the command was executed
- A configuration change sequence number

- Parser return codes for the command

You can display the information from the configuration log by using the **show archive log config** command, with the exception of the parser return codes, that are used for an internal purpose only.

Displaying Configuration Log Entries

To display the configuration log entries, the configuration change logging provides the **show archive log config all** command.

Procedure

Step 1 switch# **show archive log config** *number* [*end-number*]

Displays the configuration log entries for all users

Example:

```
switch# show archive log config all

NDEX   LINE                    USER          LOGGED COMMAND
1       console0                user01       | logging console 1
2       console0                user01       | logging monitor 2
3       console0                user02       | system default switchport shutdown
4       console0                user02       | interface mgmt0
5       console0                user02       |  no shutdown
```

Step 2 switch# **show archive log config user username**

Displays the configuration log entries for the specified username.

Example:

The following example displays the configuration log entries for a specified username.

```
switch# show archive log config user user02

INDEX   LINE                    USER          LOGGED COMMAND
3       console0                user02       | system default switchport shutdown
4       console0                user02       | interface mgmt0
5       console0                user02       |  no shutdown
```

Step 3 switch# **show archive log config user username first-index start-number [last-index end-number]**

Displays the configuration log entries by the index numbers. If you specify a number for the optional last-index, all the log entries with the index numbers in the range from the value entered for the start-number through the end-number for the specified user are displayed.

Example:

The following example displays the configuration log entry numbers 4 and 5 for a user with the username, user02. The range for the first-index and last-index is 1 to 2000000000.

```
switch# show archive log config user user02  first-index 4 last-index 5
Last Log cleared/wrapped time is : Wed Oct 19 00:53:08 2016


INDEX   LINE                    USER          LOGGED COMMAND
```

```
4       console0                user02       | interface mgmt0
5       console0                user02       |  no shutdown
```

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.

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

Examples of Working with Configuration Files

This section includes examples of working with configuration files.

Copying Configuration Files

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

Backing Up Configuration Files

This example shows how to back up the startup configuration to the bootflash: file system (ASCII file):

```
switch# copy startup-config bootflash:my-config
```

This example shows how to back up the startup configuration to the TFTP server (ASCII file):

```
switch# copy startup-config tftp://172.16.10.100/my-config
```

This example shows how to back up the running configuration to the bootflash: file system (ASCII file):

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

Rolling Back to a Previous Configuration

To roll back your configuration to a snapshot copy of a previously saved configuration, you need to perform the following steps:

1. Clear the current running image with the **write erase** command.
2. Restart the device with the **reload** command.

Note By default, the **reload** command reloads the device from a binary version of the startup configuration.

Beginning with Cisco NX-OS 6.2(2), you can use the **reload ascii** command to copy an ASCII version of the configuration to the start up configuration when reloading the device.

3. Copy the previously saved configuration file to the running configuration with the **copy** *configuration_file* **running-configuration** command.
4. Copy the running configuration to the start-up configuration with the **copy running-config startup-config** command.

Additional References for Configuration Files

This section includes additional information related to managing configuration files.

Related Documents for Configuration Files

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	