Cisco Nexus 3548 Switch NX-OS Fundamentals Configuration Guide, Release 10.2(x)

First Published: 2021-08-24

CHAPTER 1

New and Changed Information

New and Changed Information

Table 1: New and Changed Features for Cisco NX-OS Release 10.2(x)

Feature	Description	Changed in Release	Where Documented
There are no new features introduced in this release.			

CHAPTER 2

Understanding the Command-Line Interface

This chapter helps you understand the command-line interface.

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* and the *Cisco NX-OS Licensing Options Guide*.

Supported Platforms

Starting with Cisco NX-OS release 7.0(3)I7(1), use the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.

Information About the CLI Prompt

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

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

You can change the default device hostname.

From the CLI prompt, you can do the following:

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

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

Command Modes

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

EXEC Command Mode

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

Global Configuration Command Mode

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

SUMMARY STEPS

1. **configure terminal**

DETAILED STEPS

	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.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**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.

SUMMARY STEPS

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

DETAILED STEPS

	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:

SUMMARY STEPS

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

DETAILED STEPS

	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**.
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.
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).

Keystokes	Description
Ctrl-Z	Ends a configuration session, and returns you to EXEC mode. When used at the end of a command line in which a valid command has been typed, the resulting configuration is first added to the running configuration file.
Up arrow key	Displays the previous command in the command history.
Down arrow key	Displays the next command in the command history.
Right arrow key Left arrow key	Moves your cursor through the command string, either forward or backward, allowing you to edit the current command.
?	Displays a list of available commands.

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:
	
	<pre>switch(config)# c<Tab>\ncallhome class-map clock cts\ncdp cli control-plane\nswitch(config)# cl<Tab>\nclass-map cli clock\nswitch(config)# cla<Tab>\nswitch(config)# class-map</pre>
	Example:
	
	<pre>switch# cd bootflash:<Tab>\nbootflash: bootflash://sup-1/\nbootflash:/// bootflash://sup-2/\nbootflash://module-5/ bootflash://sup-active/\nbootflash://module-6/ bootflash://sup-local/</pre>
	Example:
	
	<pre>switch# cd bootflash://mo<Tab>\nbootflash://module-5/ bootflash://module-6/cv\nswitch# cd bootflash://module-</pre>

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

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

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

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

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

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

Identifying Your Location in the Command Hierarchy

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

SUMMARY STEPS

1. **where detail**

DETAILED STEPS

	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*).

SUMMARY STEPS

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

DETAILED STEPS

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

Configuring Persistent CLI Variables

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose

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

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

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

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

Command Aliases

This section provides information about command aliases.

About Command Aliases

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

Command alias support has the following characteristics:

- Command aliases are global for all user sessions.

- Command aliases persist across reboots if you save them to the startup configuration.

- Command alias translation always takes precedence over any keyword in any configuration mode or submode.

- Command alias configuration takes effect for other user sessions immediately.

- The Cisco NX-OS software provides one default alias, **alias**, which is the equivalent to the **show cli alias** command that displays all user-defined aliases.

- You cannot delete or change the default command alias **alias**.

- You can nest aliases to a maximum depth of 1. One command alias can refer to another command alias that must refer to a valid command, not to another command alias.

- A command alias always replaces the first command keyword on the command line.

- You can define command aliases for commands in any command mode.

- If you reference a CLI variable in a command alias, the current value of the variable appears in the alias, not the variable reference.

- You can use command aliases for **show** command searching and filtering.

Defining Command Aliases

You can define command aliases for commonly used commands.

SUMMARY STEPS

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

DETAILED STEPS

	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.

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

Configuring Command Aliases for a User Session

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

SUMMARY STEPS

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

DETAILED STEPS

	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.

SUMMARY STEPS

1. **run-script** [**bootflash:** | **slot0:** | **volatile:**]*filename*

DETAILED STEPS

	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.
nnn	Displays the corresponding ASCII octal character.

SUMMARY STEPS

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

DETAILED STEPS

	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.

SUMMARY STEPS

1. **sleep** *seconds*

DETAILED STEPS

	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.

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

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

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

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

Example Outputs	Description

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

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

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

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

Understanding Regular Expressions

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

Special Characters

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

This table lists the keyboard characters that have special meanings.

Table 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.

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

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

\$ _ \+

Multiple-Character Patterns

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

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

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

Anchoring

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

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

Table 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
  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.
include *string* **Example:** `show interface brief	include up`	Displays all lines that include the search string. The search string is case sensitive.
last [*lines*] **Example:** `show logging logfile	last 50`	Displays the end of the output for the number of lines specified. The default number of lines is 10.
no-more **Example:** `show interface brief	no-more`	Displays all the output without stopping at the end of the screen with the `--More--` prompt.

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

diff Utility

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

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

This table describes the keywords for the diff utility.

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

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

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

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

grep and egrep Utilities

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

The grep and egrep syntax is as follows:

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

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

Table 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.

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

Searching and Filtering from the --More-- Prompt

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

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

Table 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.

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

Using the Command History

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

Recalling a Command

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

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

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

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

Controlling CLI History Recall

You can control the commands that you recall from the CLI history using the **Ctrl-P** and **Ctrl-N** keystroke shortcuts. Cisco NX-OS software recalls all commands from the current command mode and higher command modes. For example, if you are working in global configuration mode, the command recall keystroke shortcuts recall both EXEC mode and global configuration mode commands. Using the **terminal history no-exec-in-config** command, you can avoid recalling EXEC mode commands when you are in a configuration mode.

SUMMARY STEPS

1. [**no**] **terminal history no-exec-in-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	[**no**] **terminal history no-exec-in-config** **Example:** `switch# terminal history no-exec-in-config`	Configures the CLI history to remove the EXEC commands when you use the recall keystroke shortcuts in a configuration mode. The default recalls EXEC commands. You can revert to the default using the **no** form of the command.

Configuring the CLI Edit Mode

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

SUMMARY STEPS

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

DETAILED STEPS

	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:

show cli history [*lines*] [**unformatted**]

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

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 configuration commands in the command history:

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

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

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

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

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

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.

SUMMARY STEPS

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

DETAILED STEPS

	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.

SUMMARY STEPS

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

DETAILED STEPS

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

Sending Commands to Modules

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

The **slot** has the following syntax:

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

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

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

This example shows how to display and filter module information:

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

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

```
switch# slot 2 quoted "show version" | diff
switch# slot 4 quoted "show version" | diff -c
*** /volatile/vsh_diff_1_root_8430_slot__quoted_show_version.old      Wed Apr 29 20:10:41
 2009
--- -    Wed Apr 29 20:10:41 2009
***************
```

```
*** 1,5 ****
! RAM 1036860 kB
! lc2
  Software
    BIOS:      version 1.10.6
    system:    version 4.2(1) [build 4.2(0.202)]
--- 1,5 ----
! RAM 516692 kB
! lc4
  Software
    BIOS:      version 1.10.6
    system:    version 4.2(1) [build 4.2(0.202)]
***************
*** 12,16 ****
  Hardware
      bootflash: 0 blocks (block size 512b)

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

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

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

BIOS Loader Prompt

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

```
loader>
```

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

Examples Using the CLI

This section includes examples of using the CLI.

Defining Command Aliases

This example shows how to define command aliases:

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

This example shows how to use a command alias:

```
switch# configure terminal
```

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

Using CLI Session Variables

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

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

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

Using the System-Defined Timestamp Variable

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

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

Usage for bootflash://sup-local
8192 bytes used
```

```
20963328 bytes free
20971520 bytes total
```

Running a Command Script

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

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

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

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

Related Documents for the CLI

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
Command Reference	*Cisco Nexus 3548 Switch NX-OS Fundamentals Command Reference*

CHAPTER 3

Configuring Terminal Settings and Sessions

This chapter describes how to configure terminal settings and sessions.

- Information About Terminal Settings and Sessions, on page 35
- Configuring the Console Port, on page 37
- Configuring Virtual Terminals , on page 38
- Configuring Modem Connections, on page 40
- Clearing Terminal Sessions, on page 44
- Displaying Terminal and Session Information, on page 44
- Default Settings for File System Parameters, on page 45
- Related Documents for Terminal Settings and Sessions, on page 45

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.

Number of characters displayed before wrapping the line
Cisco Nexus 3548 Switch NX-OS Fundamentals Configuration Guide, Release 10.2(x)

35

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.

SUMMARY STEPS

1. **configure terminal**
2. **line console**
3. **databits** *bits*
4. **exec-timeout** *minutes*
5. **parity** {**even** | **none** | **odd**}
6. **speed** {**300** | **1200** | **2400** | **4800** | **9600** | **38400** | **57600** | **115200**}
7. **stopbits** {**1** | **2**}
8. **exit**
9. (Optional) **show line console**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

	Command or Action	Purpose
Step 4	**exec-timeout** *minutes* **Example:** `switch(config-console)# exec-timeout 30`	Configures the timeout for an inactive session. The range is from 0 to 525600 minutes (8760 hours). A value of 0 minutes disables the session timeout. The default is 30 minutes.
Step 5	**parity** {**even** | **none** | **odd**} **Example:** `switch(config-console)# parity even`	Configures the parity. The default is **none**.
Step 6	**speed** {**300** | **1200** | **2400** | **4800** | **9600** | **38400** | **57600** | **115200**} **Example:** `switch(config-console)# speed 115200`	Configures the transmit and receive speed. The default is **9600**.
Step 7	**stopbits** {**1** | **2**} **Example:** `switch(config-console)# stopbits 2`	Configures the stop bits. The default is **1**.

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

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

SUMMARY STEPS

1. **configure terminal**
2. **line vty**
3. **exec-timeout** *minutes*

4. **exit**

5. (Optional) **show running-config all | begin vty**

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

DETAILED STEPS

	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.

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.

SUMMARY STEPS

1. **configure terminal**

2. **line vty**

3. **session-limit** *sessions*

4. **exit**

5. (Optional) **show running-config all | being vty**

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

DETAILED STEPS

	Command or Action	Purpose

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

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

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

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

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

Configuring Modem Connections

You can connect a modem to 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.

SUMMARY STEPS

1. **configure terminal**
2. **line console**

3. **modem in**
4. **exit**
5. (Optional) **show line**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**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.

SUMMARY STEPS

1. **configure terminal**
2. **line console**
3. **modem init-string default**
4. **exit**
5. (Optional) **show line**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**

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

Step 2	**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.

SUMMARY STEPS

1. **configure terminal**
2. **line console**
3. **modem set-string user-input** *string*
4. **modem init-string user-input**
5. **exit**
6. (Optional) **show line**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**

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

	Command or Action	**Purpose**
Step 2	**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 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.

SUMMARY STEPS

1.   **modem connect line console**}

DETAILED STEPS

	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.

SUMMARY STEPS

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

DETAILED STEPS

	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

Parameter	Default
Default filesystem	bootflash:

Related Documents for Terminal Settings and Sessions

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
Command Reference	*Cisco Nexus 3548 Switch NX-OS Fundamentals Command Reference*

CHAPTER **4**

Basic Device Management

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

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.

Guidelines and Limitations for Basic Device Management

The following are guidelines and limitations for basic device management:

- The **show tech-support libsdk** command and the **show tech-support mtc-usd** command are not supported. Instead use the **show tech-support module 1** command. (NX-OS 7.0(3)I7(2) and later releases.)

Changing the Device Hostname

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

SUMMARY STEPS

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

DETAILED STEPS

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

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

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 63 characters. The default name is switch.
		
		Note The **switchname** command performs the same function as the **hostname** command.

Step 3	**exit**
	
	Example:
	
	`Engineering2(config)# exit`
	`Engineering2#`
		Exits global configuration mode.

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

Configuring the MOTD Banner

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose

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

	Command or Action	Purpose

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.

SUMMARY STEPS

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

DETAILED STEPS

	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.

	Command or Action	Purpose
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.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** ```switch# configure terminal switch(config)#```	Enters global configuration mode.
Step 2	**clock summer-time** *zone-name start-week start-day start-month start-time end-week end-day end-month end-time offset-minutes* **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**.

	Command or Action	Purpose
		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.

SUMMARY STEPS

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

DETAILED STEPS

	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**.

	Command or Action	Purpose
		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.

Setting the Clock Manager

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

SUMMARY STEPS

1. **clock protocol** *protocol* **vdc** *vdc-num*
2. (Optional) **show run clock_manager**

DETAILED STEPS

	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. **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:**	Displays the configuration of the clock manager.

Command or Action	Purpose
`#show run clock_manager`	

Managing Users

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

Displaying Information about the User Sessions

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

SUMMARY STEPS

1. **show users**

DETAILED STEPS

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

Sending a Message to Users

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

SUMMARY STEPS

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

DETAILED STEPS

	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

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
Command Reference	*Cisco Nexus 3548 Switch NX-OS Fundamentals Command Reference*

CHAPTER 5

Using PowerOn Auto Provisioning

This chapter contains the following sections:

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.

Network Requirements for POAP

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
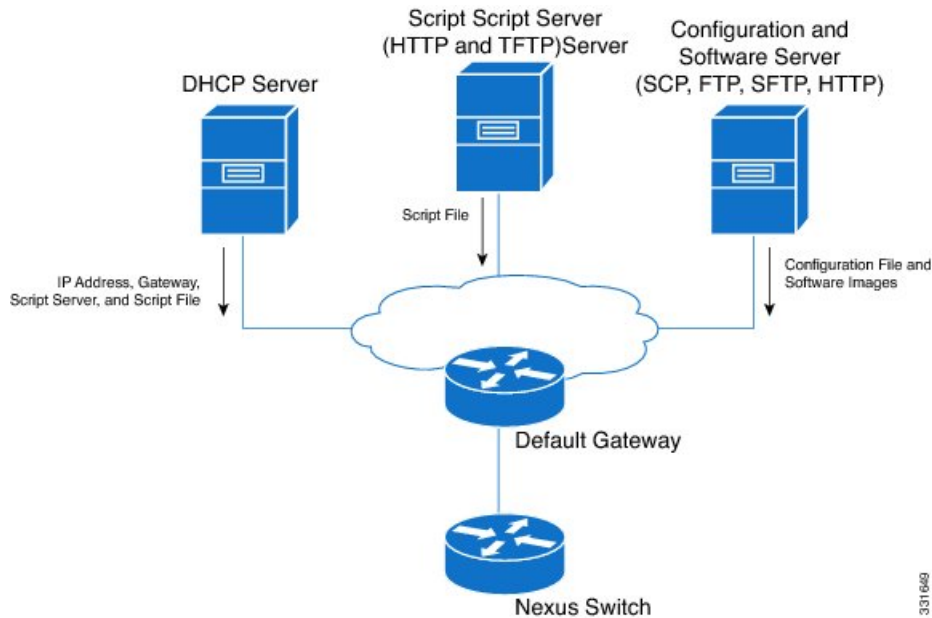
POAP Configuration Script

The reference script supplied by Cisco supports the following functionality:

• Downloads the software image (system and kickstart images) if the files do not already exist on the switch. The software image is installed on the switch and is used at the next reboot.

• Schedules the downloaded configuration to be applied at the next switch reboot.

• Stores the configuration as the startup-configuration.

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

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

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
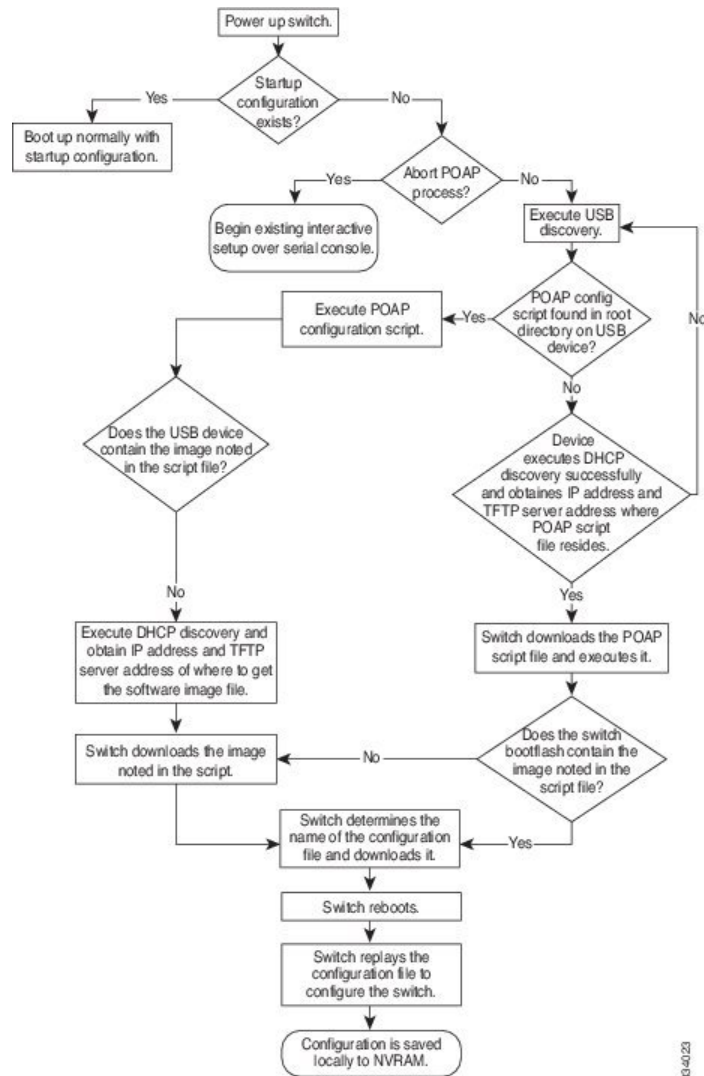
Power-Up Phase

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 (either the Python script file, poap_script.py, or the Tcl script file, poap_script.tcl), 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
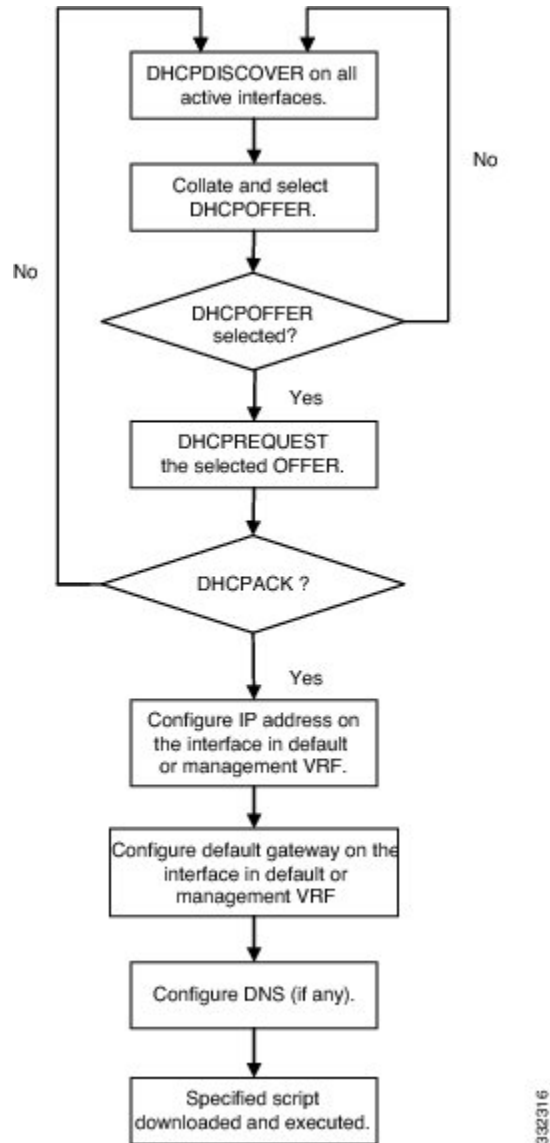
Script Execution Phase

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 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.

- Starting from Cisco NX-OS Release 7.0(3)I7(8) you can use skip_multi_level option in POAP and perform a single_image upgrade.

- 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.

- 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.

- DHCP for NX-OS will be successful, if the DHCP response is set to IP address 255.255.255.255. Since not all the DHCP server including IOS DHCP server sends the DHCP responses to 255.255.255.255, NX-OS is unable to get an IP address as a result POAP does not succeed.

- Beginning with NX-OS 7.0(3)I7(4), RFC 3004 (User Class Option for DHCP) supports POAP over IPv6 on Nexus 3000 Series switches. However, POAP over IPv6 is not supported on Nexus 3500 switches.

- 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

SUMMARY STEPS

1. Modify the basic configuration script provided by Cisco or create your own script.
2. (Optional) Put the POAP configuration script and any other desired software image and switch configuration files on a USB device accessible to the switch.
3. Deploy a DHCP server and configure it with the interface, gateway, and TFTP server IP addresses and a bootfile with the path and name of the configuration script file. (This information is provided to the switch when it first boots.)
4. Deploy a TFTP or HTTP server to host the configuration script.
5. Deploy one or more servers to host the software images and configuration files.

DETAILED STEPS

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

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

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

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

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

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

Configuring a Switch Using POAP

Before you begin

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

SUMMARY STEPS

1. Install the switch in the network.
2. Power on the switch.
3. (Optional) If you want to exit POAP mode and enter the normal interactive setup script, enter **y** (yes).

DETAILED STEPS

Step 1 Install the switch in the network.

Step 2 Power on the switch.

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

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

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

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

What to do next

Verify the configuration.

Verifying the Device Configuration

To verify the configuration, 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

C H A P T E R 6

Using the Device File Systems, Directories, and Files

This chapter describes how to use your device file systems, directories, and files.

Information About Device File Systems, Directories, Files, and External Storage Devices

This section describes the file systems, directories, files, and support provided to the external storage devices on the Cisco NX-OS devices.

File Systems

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

✎

Note The default *filesystem* parameter is bootflash:.

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

Table 16: File System Components

File System Name	Module	Description
bootflash	sup-active sup-local	Internal CompactFlash memory located on an active supervisor module. Used for storing image files, configuration files, and other miscellaneous files. The initial default directory is bootflash.
	sup-standby sup-remote	Internal CompactFlash memory located on a standby supervisor module. Used for storing image files, configuration files, and other miscellaneous files.
volatile	—	Volatile random-access memory (VRAM) located on a supervisor module. Used for temporary or pending changes.
log	—	Memory on an active supervisor module. Used for storing file statistics logs.
system	—	Memory on a supervisor module. Used for storing the running configuration file.
debug	—	Memory on a supervisor module. Used for storing the debug logs.

Directories

You can create directories on bootflash: and external flash memory (slot0:, usb1:, and usb2:). You can create, store, and access files from directories.

Files

You can create and access files from bootflash:, volatile:, slot0:, usb1:, and usb2: file systems. You can only access files from the system: file system. Use the debug: file system to store the debug log files specified using the **debug logfile** command.

You can download files, such as system image files, from remote servers using FTP, Secure Copy Protocol (SCP), Secure File Transfer Protocol (SFTP), and TFTP. You can also copy files from an external server to your device because your device can act as an SCP server.

Working with Directories

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

Identifying the Current Directory

You can display the directory name of your current directory.

SUMMARY STEPS

    1. pwd

DETAILED STEPS

	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:.

SUMMARY STEPS

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

DETAILED STEPS

	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.

SUMMARY STEPS

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

DETAILED STEPS

	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.

SUMMARY STEPS

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

DETAILED STEPS

	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.

SUMMARY STEPS

1. (Optional) **pwd**

    2. (Optional) **dir** [*filesystem* **:**[*//module/*][*directory*]]

    3. **rmdir** [*filesystem* **:**[*//module/*]]*directory*

DETAILED STEPS

	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 the Directories on a Standby Supervisor Module

You can access all the file systems on a standby supervisor module (remote) from a session on an active supervisor module. This feature is useful when copying files to the active supervisor module that requires similar files to exist, as in the standby supervisor module.

To access the file systems on the standby supervisor module from a session on the active supervisor module, specify the standby supervisor module in the path to the file using either the *filesystem***://sup-remote/** command, or the *filesystem***://sup-standby/** command.

Working with Files

This section describes how to work with files on a Cisco NX-OS device.

Moving Files

You can move a file from one directory to another directory.

⚠️

Caution If a file with the same name already exists in the destination directory, that file is overwritten by the moved file.

You can use the **move** command to rename a file by moving the file within the same directory.

SUMMARY STEPS

    1. (Optional) **pwd**

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

DETAILED STEPS

	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.

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

SUMMARY STEPS

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

DETAILED STEPS

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

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

Deleting Files

You can delete a file from a directory.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **dir** [*filesystem***:**[*//module/*][*directory*]] **Example:** `switch# dir bootflash`	Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 2	**delete** {*filesystem***:**[*//module/*][*directory/*] | *directory/*}*filename* **Example:** `switch# delete test old_tests/test1`	Deletes a file. The file system, module, and directory names are case sensitive. The *source-filename* argument is case sensitive. **Caution** If you specify a directory, the **delete** command deletes the entire directory and all its contents.

Displaying File Contents

You can display the contents of a file.

SUMMARY STEPS

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

DETAILED STEPS

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

Displaying File Checksums

You can display checksums to check the file integrity.

SUMMARY STEPS

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

DETAILED STEPS

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

Compressing and Uncompressing Files

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	(Optional) **dir** [*filesystem***:**[*//module/*]*directory*]]	Displays the contents of the current directory. The file system and directory name are case sensitive.
	Example:	
	`switch# dir bootflash:`	
Step 2	**gzip** [*filesystem***:**[*//module/*][*directory/*]	*directory/*]*filename*	Compresses a file. After the file is compressed, it has a .gz suffix.
	Example:	
	`switch# gzip show_tech`	

	Command or Action	Purpose
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.

SUMMARY STEPS

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

DETAILED STEPS

	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. You can also specify the format for the command output.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **terminal redirection-mode** {**ascii** | **zipped**} **Example:** `switch# terminal redirection-mode zipped`	Sets the redirection mode for the **show** command output for the user session. The default mode is **ascii**.
Step 2	*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.

SUMMARY STEPS

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

DETAILED STEPS

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

Working with Archive Files

The Cisco NX-OS software supports archive files. Besides creating an archive file, you can append files to, extract files from, and list the files in an archive file.

Creating an Archive Files

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

- bzip2
- gzip
- Uncompressed

The default is gzip.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**tar create** {**bootflash:** | **volatile:**}*archive-filename* [**absolute**] [**bz2-compress**] [**gz-compress**] [**remove**] [**uncompressed**] [**verbose**] *filename-list*	Creates an archive file and adds files to it. The filename is alphanumeric, not case sensitive, and has a maximum length of 240 characters.
		The **absolute** keyword specifies that the leading backslash characters (\\) should not be removed from the names of the files added to the archive file. By default, the leading backslash characters are removed.
		The **bz2-compress**, **gz-compress**, and **uncompressed** keywords determine the compression utility used when files are added, or later appended, to the archive and the decompression utility to use when extracting the files. If you do not specify an extension for the archive file, the defaults are as follows:
		• For **bz2-compress**, the extension is .tar.bz2.
		• For **gz-compress**, the extension is .tar.gz.
		• For **uncompressed**, the extension is .tar.
		The **remove** keyword specifies that the Cisco NX-OS software should delete the files from the file system after adding them to the archive. By default, the files are not deleted.
		The **verbose** keyword specifies that the Cisco NX-OS software should list the files as they are added to the archive. By default, the files are listed as they are added.

Example

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

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

Appending Files to an Archive File

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

Before you begin

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

SUMMARY STEPS

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

DETAILED STEPS

		Command or Action	Purpose
Step 1		**tar append** {**bootflash:**	**volatile:**}*archive-filename* [**absolute**] [**remove**] [**verbose**] *filename-list*	Adds files to an existing archive file. The archive filename is not case sensitive.
			The **absolute** keyword specifies that the leading backslash characters (\) should not be removed from the names of the files added to the archive file. By default, the leading backslash characters are removed.
			The **remove** keyword specifies that the Cisco NX-OS software should delete the files from the filesystem after adding them to the archive. By default, the files are not deleted.
			The **verbose** keyword specifies that the Cisco NX-OS software should list the files as they are added to the archive. By default, the files are listed as they are added.

Example

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

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

Extracting Files from an Archive File

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

Before you begin

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

SUMMARY STEPS

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

DETAILED STEPS

		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.

Command or Action	Purpose
	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

> **Note** The archive filename is not case sensitive.

To display the file names in an archive file, run the following command:

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

Example:

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

Examples of Using a File System

This section includes examples of using a file system on a Cisco NX-OS device.

Accessing Directories on a Standby Supervisor Module

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

```
switch# dir bootflash://sup-remote
   12198912     Aug 27 16:29:18 2003  m9500-sf1ek9-kickstart-mzg.1.3.0.39a.bin
    1864931     Apr 29 12:41:59 2003  dplug2
      12288     Apr 18 20:23:11 2003  lost+found/
   12097024     Nov 21 16:34:18 2003  m9500-sf1ek9-kickstart-mz.1.3.1.1.bin
   41574014     Nov 21 16:34:47 2003  m9500-sf1ek9-mz.1.3.1.1.bin

Usage for bootflash://sup-remote
  67747169 bytes used
```

```
       116812447 bytes free
       184559616 bytes total
```

This example shows how to delete a file on a standby supervisor module:

```
switch# delete bootflash://sup-remote/aOldConfig.txt
```

Moving Files

This example shows how to move a file on an external flash device:

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

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

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

Copying Files

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

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

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

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

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

```
switch# copy bootflash:system_image bootflash://sup-2/system_image
```

Note   You can also use the **copy** command to upload and download files from the slot0: or bootflash: file system to or from an FTP, TFTP, SFTP, or SCP server.

Deleting a Directory

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

Before you begin

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

SUMMARY STEPS

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

DETAILED STEPS

	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

Parameter	Default
Default filesystem	bootflash:

Additional References for File Systems

This section includes additional information related to the file systems.

Related Documents for File Systems

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

Working with Configuration Files

This chapter describes how to work with your device configuration files.

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.

Guidelines and Limitations for Configuration Files

Configuration file guidelines and limitations are as follows:

Managing Configuration Files

This section describes how to manage configuration files.

Saving the Running Configuration to the Startup Configuration

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

SUMMARY STEPS

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

DETAILED STEPS

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

Copying a Configuration File to a Remote Server

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**copy running-config** *scheme***://***server***/[***url* /**]***filename* **Example:**	Copies the running-configuration file to a remote server.

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

Example

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

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

Downloading the Running Configuration From a Remote Server

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

Before you begin

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

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

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

Downloading the Startup Configuration From a Remote Server

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

⚠️

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

Before you begin

Log in to a session on the console port.

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

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

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

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

SUMMARY STEPS

1.   **write erase**

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

DETAILED STEPS

	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:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.
		
		The *server*, *url*, and *filename* arguments are case sensitive.

Step 4	**copy running-config startup-config**
	
	Example:
	
	`switch# copy running-config`
	`startup-config`
		Saves the running configuration file to the startup configuration file.

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

Copying Configuration Files to an External Flash Memory Device

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

Before you begin

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

SUMMARY STEPS

1. (Optional) **dir** {**slot0:** | **usb1:** | **usb2:**}[*directory*/]
2. **copy running-config** {**slot0:** | **usb1:** | **usb2:**}[*directory*/]*filename*
3. **copy startup-config** {**slot0:** | **usb1:** | **usb2:**}[*directory*/]*filename*

DETAILED STEPS

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

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

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

Copying the Running Configuration from an External Flash Memory Device

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

Before you begin

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

SUMMARY STEPS

1. (Optional) **dir** {**slot0:** | **usb1:** | **usb2:**}[*directory*/]
2. **copy** {**slot0:** | **usb1:** | **usb2:**}[*directory*/]*filename* **running-config**
3. (Optional) **show running-config**
4. (Optional) **copy running-config startup-config**
5. (Optional) **show startup-config**

DETAILED STEPS

	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** {**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:** switch# copy running-config startup-config	Copies the running configuration to the startup configuration.
Step 5	(Optional) **show startup-config** **Example:** switch# show startup-config	Displays the startup configuration.

Copying the Startup Configuration from an External Flash Memory Device

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

Before you begin

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

SUMMARY STEPS

1. (Optional) **dir** {**slot0:** \| **usb1:** \| **usb2:**}[*directory*/]
2. **copy** {**slot0:** \| **usb1:** \| **usb2:**}[*directory* /]*filename* **startup-config**
3. (Optional) **show startup-config**

DETAILED STEPS

	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.

	Command or Action	**Purpose**
Step 3	(Optional) **show startup-config** **Example:** `switch# show startup-config`	Displays the startup configuration.

Copying Configuration Files to an Internal File System

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

SUMMARY STEPS

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

DETAILED STEPS

	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.

Rolling Back to a Previous Configuration

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

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

SUMMARY STEPS

1. **write erase**
2. **reload**
3. **copy** *configuration_file* **running-configuration**

4. copy running-config startup-config

DETAILED STEPS

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

Step 3	**copy** *configuration_file* **running-configuration**
	
	Example:
	`switch# copy bootflash:start-config.bak`
	`running-configuration`
		Copies a previously saved configuration file to the running configuration.
		
		Note The *configuration_file* filename argument is case sensitive.

Step 4	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the start-up configuration.

Removing the Configuration for a Missing Module

When you remove an I/O module from the chassis, you can also remove the configuration for that module from the running configuration.

✏️

Note You can only remove the configuration for an empty slot in the chassis.

Before you begin

Remove the I/O module from the chassis.

SUMMARY STEPS

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

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

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	(Optional) **show hardware** **Example:** `switch# show hardware`	Displays the installed hardware for the device.
Step 2	**purge module** *slot* **running-config** **Example:** `switch# purge module 3 running-config`	Removes the configuration for a missing module from the running configuration.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Erasing a Configuration

You can erase the configuration on your device to return to the factory defaults.

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

- Startup

- Boot

- Debug

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

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

    - Address
    - Subnet mask

To remove the boot variable definitions follow step-1 and step-2.

To remove the boot variables, running configuration, and the IP configuration on the management interface follow step-3 to step-5.

SUMMARY STEPS

1. **write erase boot**
2. **reload**
3. **write erase**
4. **write erase boot**
5. **reload**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**write erase boot** **Example:** `switch# write erase boot`	Erases the boot variable definitions.
Step 2	**reload** **Example:** `switch# reload`	Restarts the device. You will be prompted to provide a kickstart and system image file for the device to boot and run. By default, the reload command reloads the device from a binary version of the startup configuration.
Step 3	**write erase** **Example:** `switch# write erase`	Erases the boot variable definitions.
Step 4	**write erase boot** **Example:** `switch# write erase boot`	Erases the boot variable definitions and the IPv4 configuration on the management interface.
Step 5	**reload** **Example:** `switch# reload`	Restarts the device. You will be prompted to provide a kickstart and system image file for the device to boot and run. By default, the reload command reloads the device from a binary version of the startup configuration.

Clearing Inactive Configurations

You can clear inactive Quality of Service (QoS) and/or access control list (ACL) configurations.

SUMMARY STEPS

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

DETAILED STEPS

	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.

	Command or Action	Purpose
		• **ipqos**—Displays any inactive configurations for qosmgr.
Step 2	**clear inactive-config** *policy* **Example:** ```# clear inactive-config qos clear qos inactive config Inactive if config for QoS manager is saved at/bootflash/qos_inactive_if_config.cfg for vdc default & for other than default vdc: /bootflash/vdc_x/qos_inactive_if_config.cfg (where x is vdc number) you can see the log file @ show inactive-if-config log```	Clears inactive configurations. The values for the *policy* argument are **qos** and **acl**. The following describes the values: • **qos**—Clears inactive QoS configurations. • **acl**— Clears inactive ACL configurations. • **acl qos**—Clears inactive ACL configurations and inactive QoS configurations.
Step 3	(Optional) **show inactive-if-config log** **Example:** ```# show inactive-if-config log```	Displays the commands that were used to clear the inactive configurations.

Verifying the Device Configuration

To verify the configuration, 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:

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

Backing Up Configuration Files

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

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

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

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

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

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

Rolling Back to a Previous Configuration

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

1. Clear the current running image with the **write erase** command.

2. Restart the device with the **reload** command.

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

Related Documents for Configuration files

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
Command Reference	*Cisco Nexus 3548 Switch NX-OS Fundamentals Command Reference*