



# Understanding the Command-Line Interface

---

- [Licensing Requirements, on page 1](#)
- [Information About the CLI Prompt, on page 1](#)
- [Command Modes, on page 2](#)
- [Configuring CLI Variables, on page 8](#)
- [Command Aliases, on page 8](#)
- [Command Scripts, on page 8](#)
- [Understanding Regular Expressions, on page 10](#)
- [Searching and Filtering show Command Output, on page 10](#)
- [Using the Command History, on page 11](#)
- [Examples Using the CLI , on page 13](#)
- [Additional References for the CLI, on page 14](#)

## 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](#).

## Information About the CLI Prompt

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

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

```
switch#
```

You can change the default device hostname.

From the CLI prompt, you can do the following:

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




---

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

---

## Command Modes

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

## 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 1: 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 2: 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.

Keystokes	Description
Tab	<p>Completes the word for you after you enter the first characters of the word and then press the Tab key. All options that match are presented.</p> <p>Use tabs to complete the following items:</p> <ul style="list-style-type: none"> <li>• Command names</li> <li>• Scheme names in the file system</li> <li>• Server names in the file system</li> <li>• Filenames in the file system</li> </ul> <p><b>Example:</b></p> <pre>switch(config)# xm&lt;Tab&gt; switch(config)# xml&lt;Tab&gt; switch(config)# xml server</pre> <p><b>Example:</b></p> <pre>switch(config)# c&lt;Tab&gt; callhome class-map clock cts cdp cli control-plane switch(config)# cl&lt;Tab&gt; class-map cli clock switch(config)# cla&lt;Tab&gt; switch(config)# class-map</pre> <p><b>Example:</b></p> <pre>switch# cd bootflash:&lt;Tab&gt; bootflash: bootflash://sup-1/ bootflash:/// bootflash://sup-2/ bootflash://module-5/ bootflash://sup-active/ bootflash://module-6/ bootflash://sup-local/</pre> <p><b>Example:</b></p> <pre>switch# cd bootflash://mo&lt;Tab&gt; bootflash://module-5/ bootflash://module-6/cv switch# 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 3: Examples of Command Abbreviations*

Command	Abbreviation
configure terminal	conf t
copy running-config startup-config	copy run start
interface ethernet 1/2	int e 1/2
show running-config	sh run

## Completing a Partial Command Name

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

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

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

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

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

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

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

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

## Identifying Your Location in the Command Hierarchy

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

### SUMMARY STEPS

1. where detail

## DETAILED STEPS

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

## Using the no Form of a Command

Almost every configuration command has a **no** form that can be used to disable a feature, revert to a default value, or remove a configuration. 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.

## Command Aliases

This section provides information about command aliases.

## Command Scripts

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

## 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 4: Context-Sensitive Help Example**

Example Outputs	Description
<pre>switch# clock ?   set HH:MM:SS Current Time switch# clock</pre>	<p>Displays the command syntax for the <b>clock</b> command in EXEC mode.</p> <p>The switch output shows that the <b>set</b> keyword is required for using the <b>clock</b> command.</p>
<pre>switch# clock set ?   WORD HH:MM:SS Current Time switch# clock set</pre>	<p>Displays the command syntax for setting the time.</p> <p>The help output shows that the current time is required for setting the clock and how to format the time.</p>
<pre>switch# clock set 13:32:00&lt;CR&gt; % Incomplete command switch#</pre>	<p>Adds the current time.</p> <p>The CLI indicates the command is incomplete.</p>
<pre>switch# &lt;Ctrl-P&gt; switch# clock set 13:32:00</pre>	<p>Displays the previous command that you entered.</p>
<pre>switch# clock set 13:32:00 ?   &lt;1-31&gt; Day of the month switch# clock set 13:32:00</pre>	<p>Displays the additional arguments for the <b>clock set</b> command.</p>
<pre>switch# clock set 13:32:00 18 ?   April      Month of the year   August     Month of the year   December   Month of the year   February   Month of the year   January     Month of the year   July       Month of the year   June       Month of the year   March      Month of the year   May        Month of the year   November   Month of the year   October    Month of the year   September  Month of the year switch# clock set 13:32:00 18</pre>	<p>Displays the additional arguments for the <b>clock set</b> command.</p>
<pre>switch# clock set 13:32:00 18 April 08&lt;CR&gt; % Invalid input detected at '^' marker.</pre>	<p>Adds the date to the clock setting.</p> <p>The CLI indicates an error with the caret symbol (^) at 08.</p>
<pre>switch# clock set 13:32:00 18 April ?   &lt;2000-2030&gt; Enter the year (no abbreviation)  switch# clock set 13:32:00 18 April</pre>	<p>Displays the correct arguments for the year.</p>
<pre>switch# clock set 13:32:00 18 April 2008&lt;CR&gt; switch#</pre>	<p>Enters the correct syntax for the <b>clock set</b> command.</p>

# Understanding Regular Expressions

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

## Searching and Filtering show Command Output

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

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

## 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 5: --More-- Prompt Commands**

Commands	Description
[lines]<space>	Displays output lines for either the specified number of lines or the current screen size.

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

## 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
<b>Step 1</b>	<p>[no] terminal dont-ask [persist]</p> <p><b>Example:</b></p> <pre>switch# terminal dont-ask</pre>	<p>Disables the CLI confirmation prompt. The <b>persist</b> keyword makes the setting persistent across sessions for the current username. The default is enabled.</p> <p>Use the <b>no</b> form of the command to enable the CLI confirmation prompts.</p>

## Setting CLI Display Colors

You can change the CLI colors to display as follows:

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

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

## SUMMARY STEPS

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

## DETAILED STEPS

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

## Sending Commands to Modules

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

The **slot** has the following syntax:

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

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

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

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

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

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

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

# Additional References for the CLI

This section includes additional information related to the CLI.