# Using the Command-Line Interface


- Information About Using the Command-Line Interface, page 1
- How to Use the CLI to Configure Features, page 6


# Information About Using the Command-Line Interface

## Command Modes

The Cisco IOS user interface is divided into many different modes. The commands available to you depend on which mode you are currently in. Enter a question mark (?) at the system prompt to obtain a list of commands available for each command mode.

You can start a CLI session through a console connection, through Telnet, a SSH, or by using the browser.

When you start a session, you begin in user mode, often called user EXEC mode. Only a limited subset of the commands are available in user EXEC mode. For example, most of the user EXEC commands are one-time commands, such as **show** commands, which show the current configuration status, and **clear** commands, which clear counters or interfaces. The user EXEC commands are not saved when the controller reboots.

To have access to all commands, you must enter privileged EXEC mode. Normally, you must enter a password to enter privileged EXEC mode. From this mode, you can enter any privileged EXEC command or enter global configuration mode.

Using the configuration modes (global, interface, and line), you can make changes to the running configuration. If you save the configuration, these commands are stored and used when the controller reboots. To access the various configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and line configuration mode.

This table describes the main command modes, how to access each one, the prompt you see in that mode, and how to exit the mode.

*Table 1: Command Mode Summary*

| Mode | Access Method | Prompt | Exit Method | About This Mode |
|---|---|---|---|---|
| User EXEC | Begin a session using Telnet, SSH, or console. | `Controller>` | Enter **logout** or **quit**. | Use this mode to<br>• Change terminal settings.<br>• Perform basic tests.<br>• Display system information. |
| Privileged EXEC | While in user EXEC mode, enter the **enable** command. | `Controller#` | Enter **disable** to exit. | Use this mode to verify commands that you have entered. Use a password to protect access to this mode.<br>Use this mode to execute privilege EXEC commands for access points. These commands are not part of the running config of the controller, they are sent to the IOS config of the access point. |
| Global configuration | While in privileged EXEC mode, enter the **configure** command. | `Controller(config)#` | To exit to privileged EXEC mode, enter **exit** or **end**, or press **Ctrl-Z**. | Use this mode to configure parameters that apply to the entire controller.<br>Use this mode to configure access point commands that are part of the running config of the controller. |
| VLAN configuration | While in global configuration mode, enter the **vlan** *vlan-id* command. | `Controller(config-vlan)#` | | |

| Mode | Access Method | Prompt | Exit Method | About This Mode |
|---|---|---|---|---|
| | | | To exit to global configuration mode, enter the **exit** command.<br>To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**. | Use this mode to configure VLAN parameters. When VTP mode is transparent, you can create extended-range VLANs (VLAN IDs greater than 1005) and save configurations in the controller startup configuration file. |
| Interface configuration | While in global configuration mode, enter the **interface** command (with a specific interface). | `Controller(config-if)#` | To exit to global configuration mode, enter **exit**.<br>To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**. | Use this mode to configure parameters for the Ethernet ports. |
| Line configuration | While in global configuration mode, specify a line with the **line vty** or **line console** command. | `Controller(config-line)#` | To exit to global configuration mode, enter **exit**.<br>To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**. | Use this mode to configure parameters for the terminal line. |

# Using the Help System

You can enter a question mark (?) at the system prompt to display a list of commands available for each command mode. You can also obtain a list of associated keywords and arguments for any command.

### SUMMARY STEPS

1. **help**
2. *abbreviated-command-entry* **?**
3. *abbreviated-command-entry* <Tab>
4. **?**
5. *command* **?**
6. *command keyword* **?**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **help**<br><br>**Example:**<br>`Controller# help` | Obtains a brief description of the help system in any command mode. |
| **Step 2** | *abbreviated-command-entry* **?**<br><br>**Example:**<br>`Controller# di?`<br>`dir disable disconnect` | Obtains a list of commands that begin with a particular character string. |
| **Step 3** | *abbreviated-command-entry* <Tab><br><br>**Example:**<br>`Controller# sh conf<tab>`<br>`Controller# show configuration` | Completes a partial command name. |
| **Step 4** | **?**<br><br>**Example:**<br>`Controller> ?` | Lists all commands available for a particular command mode. |
| **Step 5** | *command* **?**<br><br>**Example:**<br>`Controller> show ?` | Lists the associated keywords for a command. |
| **Step 6** | *command keyword* **?**<br><br>**Example:**<br>`Controller(config)# cdp holdtime ?`<br>`<10-255> Length of time (in sec) that receiver must keep this packet` | Lists the associated arguments for a keyword. |

# Understanding Abbreviated Commands

You need to enter only enough characters for the controller to recognize the command as unique.

This example shows how to enter the **show configuration** privileged EXEC command in an abbreviated form:

```
Controller# show conf
```

# No and Default Forms of Commands

Almost every configuration command also has a **no** form. In general, use the **no** form to disable a feature or function or reverse the action of a command. For example, the **no shutdown** interface configuration command reverses the shutdown of an interface. Use the command without the keyword **no** to reenable a disabled feature or to enable a feature that is disabled by default.

Configuration commands can also have a **default** form. The **default** form of a command returns the command setting to its default. Most commands are disabled by default, so the **default** form is the same as the **no** form. However, some commands are enabled by default and have variables set to certain default values. In these cases, the **default** command enables the command and sets variables to their default values.

# CLI Error Messages

This table lists some error messages that you might encounter while using the CLI to configure your controller.

***Table 2: Common CLI Error Messages***

| Error Message | Meaning | How to Get Help |
|---|---|---|
| `% Ambiguous command: "show con"` | You did not enter enough characters for your controller to recognize the command. | Reenter the command followed by a question mark (?) without any space between the command and the question mark.<br>The possible keywords that you can enter with the command appear. |
| `% Incomplete command.` | You did not enter all of the keywords or values required by this command. | Reenter the command followed by a question mark (?) with a space between the command and the question mark.<br>The possible keywords that you can enter with the command appear. |
| `% Invalid input detected at '^' marker.` | You entered the command incorrectly. The caret (^) marks the point of the error. | Enter a question mark (?) to display all of the commands that are available in this command mode.<br>The possible keywords that you can enter with the command appear. |

# Configuration Logging

You can log and view changes to the controller configuration. You can use the Configuration Change Logging and Notification feature to track changes on a per-session and per-user basis. The logger tracks each configuration command that is applied, the user who entered the command, the time that the command was entered, and the parser return code for the command. This feature includes a mechanism for asynchronous

notification to registered applications whenever the configuration changes. You can choose to have the notifications sent to the syslog.

**Note** Only CLI or HTTP changes are logged.

# How to Use the CLI to Configure Features

## Configuring the Command History

The software provides a history or record of commands that you have entered. The command history feature is particularly useful for recalling long or complex commands or entries, including access lists. You can customize this feature to suit your needs.

### Changing the Command History Buffer Size

By default, the controller records ten command lines in its history buffer. You can alter this number for a current terminal session or for all sessions on a particular line. This procedure is optional.

#### SUMMARY STEPS

1. **terminal history** [**size** *number-of-lines*]

#### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **terminal history** [**size** *number-of-lines*]<br><br>**Example:**<br>`Controller# terminal history size 200` | Changes the number of command lines that the controller records during the current terminal session in privileged EXEC mode. You can configure the size from 0 to 256. |

### Recalling Commands

To recall commands from the history buffer, perform one of the actions listed in this table. These actions are optional.

**Note** The arrow keys function only on ANSI-compatible terminals such as VT100s.

### SUMMARY STEPS

1. **Ctrl-P** or use the **up arrow** key
2. **Ctrl-N** or use the **down arrow** key
3. **show history**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **Ctrl-P** or use the **up arrow** key | Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands. |
| **Step 2** | **Ctrl-N** or use the **down arrow** key | Returns to more recent commands in the history buffer after recalling commands with **Ctrl-P** or the up arrow key. Repeat the key sequence to recall successively more recent commands. |
| **Step 3** | **show history**<br><br>**Example:**<br>`Controller# show history` | Lists the last several commands that you just entered in privileged EXEC mode. The number of commands that appear is controlled by the setting of the **terminal history** global configuration command and the **history** line configuration command. |

## Disabling the Command History Feature

The command history feature is automatically enabled. You can disable it for the current terminal session or for the command line. This procedure is optional.

### SUMMARY STEPS

1. **terminal no history**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **terminal no history**<br><br>**Example:**<br>`Controller# terminal no history` | Disables the feature during the current terminal session in privileged EXEC mode. |

# Enabling and Disabling Editing Features

Although enhanced editing mode is automatically enabled, you can disable it and reenable it.

## SUMMARY STEPS

1. terminal editing
2. terminal no editing

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **terminal editing**<br><br>**Example:**<br>`Controller# terminal editing` | Reenables the enhanced editing mode for the current terminal session in privileged EXEC mode. |
| Step 2 | **terminal no editing**<br><br>**Example:**<br>`Controller# terminal no editing` | Disables the enhanced editing mode for the current terminal session in privileged EXEC mode. |

## Editing Commands Through Keystrokes

The keystrokes help you to edit the command lines. These keystrokes are optional.

**Note** The arrow keys function only on ANSI-compatible terminals such as VT100s.

***Table 3: Editing Commands***

| Editing Commands | Description |
|---|---|
| **Ctrl-B** or use the **left arrow** key | Moves the cursor back one character. |
| **Ctrl-F** or use the **right arrow** key | Moves the cursor forward one character. |
| **Ctrl-A** | Moves the cursor to the beginning of the command line. |
| **Ctrl-E** | Moves the cursor to the end of the command line. |
| **Esc B** | Moves the cursor back one word. |
| **Esc F** | Moves the cursor forward one word. |
| **Ctrl-T** | Transposes the character to the left of the cursor with the character located at the cursor. |
| **Delete** or **Backspace** key | Erases the character to the left of the cursor. |
| **Ctrl-D** | Deletes the character at the cursor. |
| **Ctrl-K** | Deletes all characters from the cursor to the end of the command line. |
| **Ctrl-U** or **Ctrl-X** | Deletes all characters from the cursor to the beginning of the command line. |
| **Ctrl-W** | Deletes the word to the left of the cursor. |
| **Esc D** | Deletes from the cursor to the end of the word. |
| **Esc C** | Capitalizes at the cursor. |
| **Esc L** | Changes the word at the cursor to lowercase. |
| **Esc U** | Capitalizes letters from the cursor to the end of the word. |

| | |
|---|---|
| **Ctrl-V** or **Esc Q** | Designates a particular keystroke as an executable command, perhaps as a shortcut. |
| **Return** key | Scrolls down a line or screen on displays that are longer than the terminal screen can display.<br>**Note** The More prompt is used for any output that has more lines than can be displayed on the terminal screen, including **show** command output. You can use the **Return** and **Space** bar keystrokes whenever you see the More prompt. |
| **Space** bar | Scrolls down one screen. |
| **Ctrl-L** or **Ctrl-R** | Redisplays the current command line if the controller suddenly sends a message to your screen. |

## Editing Command Lines That Wrap

You can use a wraparound feature for commands that extend beyond a single line on the screen. When the cursor reaches the right margin, the command line shifts ten spaces to the left. You cannot see the first ten characters of the line, but you can scroll back and check the syntax at the beginning of the command. The keystroke actions are optional.

To scroll back to the beginning of the command entry, press **Ctrl-B** or the left arrow key repeatedly. You can also press **Ctrl-A** to immediately move to the beginning of the line.

**Note** The arrow keys function only on ANSI-compatible terminals such as VT100s.

The following example shows how to wrap a command line that extends beyond a single line on the screen.

### SUMMARY STEPS

1. **access-list**
2. **Ctrl-A**
3. **Return** key

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **access-list**<br><br>**Example:**<br><br>`Controller(config)# access-list 101 permit` | Displays the global configuration command entry that extends beyond one line.<br><br>When the cursor first reaches the end of the line, the line is shifted ten spaces to the left and redisplayed. The dollar sign ($) shows that the |

| | Command or Action | Purpose |
|---|---|---|
| | **tcp 10.15.22.25 255.255.255.0 10.15.22.35**<br>Controller(config)# $ **101 permit tcp 10.15.22.25 255.255.255.0 10.15.22.35 255.25**<br>Controller(config)# $**t tcp 10.15.22.25 255.255.255.0 131.108.1.20 255.255.255.0 eq**<br>Controller(config)# $**15.22.25 255.255.255.0 10.15.22.35 255.255.255.0 eq 45** | line has been scrolled to the left. Each time the cursor reaches the end of the line, the line is again shifted ten spaces to the left. |
| **Step 2** | **Ctrl-A**<br><br>**Example:**<br>Controller(config)# **access-list 101 permit tcp 10.15.22.25 255.255.255.0 10.15.2$** | Checks the complete syntax.<br>The dollar sign ($) appears at the end of the line to show that the line has been scrolled to the right. |
| **Step 3** | **Return** key | Execute the commands.<br>The software assumes that you have a terminal screen that is 80 columns wide. If you have a different width, use the **terminal width** privileged EXEC command to set the width of your terminal.<br>Use line wrapping with the command history feature to recall and modify previous complex command entries. |

# Searching and Filtering Output of show and more Commands

You can search and filter the output for **show** and **more** commands. This is useful when you need to sort through large amounts of output or if you want to exclude output that you do not need to see. Using these commands is optional.

## SUMMARY STEPS

1. {**show** | **more**} *command* | {**begin** | **include** | **exclude**} *regular-expression*

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | {**show** | **more**} *command* | {**begin** | **include** | **exclude**} *regular-expression*<br><br>**Example:**<br>Controller# **show interfaces | include protocol**<br>Vlan1 is up, line protocol is up<br>Vlan10 is up, line protocol is down<br>GigabitEthernet1/0/1 is up, line protocol is down<br>GigabitEthernet1/0/2 is up, line protocol is up | Searches and filters the output.<br>Expressions are case sensitive. For example, if you enter **| exclude output**, the lines that contain **output** are not displayed, but the lines that contain **output** appear. |

# Accessing the CLI

You can access the CLI through a console connection, through Telnet, a SSH, or by using the browser.

# Accessing the CLI Through a Console Connection or Through Telnet

Before you can access the CLI, you must connect a terminal or a PC to the controller console or connect a PC to the Ethernet management port and then power on the controller, as described in the hardware installation guide that shipped with your controller.

If your controller is already configured, you can access the CLI through a local console connection or through a remote Telnet session, but your controller must first be configured for this type of access.

You can use one of these methods to establish a connection with the controller:

- Connect the controller console port to a management station or dial-up modem, or connect the Ethernet management port to a PC. For information about connecting to the console or Ethernet management port, see the controller hardware installation guide.
- Use any Telnet TCP/IP or encrypted Secure Shell (SSH) package from a remote management station. The controller must have network connectivity with the Telnet or SSH client, and the controller must have an enable secret password configured.
    - The controller supports up to 16 simultaneous Telnet sessions. Changes made by one Telnet user are reflected in all other Telnet sessions.
    - The controller supports up to five simultaneous secure SSH sessions.

  After you connect through the console port, through the Ethernet management port, through a Telnet session or through an SSH session, the user EXEC prompt appears on the management station.