



Using the Command-Line Interface

This chapter describes the IOS command-line interface (CLI) that you can use to configure your bridge. It contains these sections:

- [IOS Command Modes, page 4-2](#)
- [Getting Help, page 4-3](#)
- [Abbreviating Commands, page 4-3](#)
- [Using no and default Forms of Commands, page 4-3](#)
- [Understanding CLI Messages, page 4-4](#)
- [Using Command History, page 4-4](#)
- [Using Editing Features, page 4-5](#)
- [Searching and Filtering Output of show and more Commands, page 4-8](#)
- [Accessing the CLI, page 4-8](#)

IOS 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.

When you start a session on the bridge, 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 bridge 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 must enter privileged EXEC mode before you can enter the 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 bridge 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.

Table 4-1 describes the main command modes, how to access each one, the prompt you see in that mode, and how to exit the mode. The examples in the table use the host name *BR*.

Table 4-1 Command Mode Summary

Mode	Access Method	Prompt	Exit Method	About This Mode
User EXEC	Begin a session with your bridge.	BR>	Enter logout or quit .	Use this mode to: <ul style="list-style-type: none"> • Change terminal settings • Perform basic tests • Display system information
Privileged EXEC	While in user EXEC mode, enter the enable command.	BR#	Enter disable to exit.	Use this mode to verify commands. Use a password to protect access to this mode.
Global configuration	While in privileged EXEC mode, enter the configure command.	BR(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 bridge.
Interface configuration	While in global configuration mode, enter the interface command (with a specific interface).	BR(config-if)#	To exit to global configuration mode, enter exit . To return to privileged EXEC mode, press Ctrl-Z or enter end .	Use this mode to configure parameters for the Ethernet and radio interfaces. The 2.4-GHz radio is radio 0, and the 5-GHz radio is radio 1.

Getting Help

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, as shown in [Table 4-2](#).

Table 4-2 Help Summary

Command	Purpose
<code>help</code>	Obtains a brief description of the help system in any command mode.
<code>abbreviated-command-entry?</code>	Obtains a list of commands that begin with a particular character string. For example: BR# di? dir disable disconnect
<code>abbreviated-command-entry<Tab></code>	Completes a partial command name. For example: BR# sh conf <tab> BR# show configuration
<code>?</code>	Lists all commands available for a particular command mode. For example: BR> ?
<code>command ?</code>	Lists the associated keywords for a command. For example: BR> show ?
<code>command keyword ?</code>	Lists the associated arguments for a keyword. For example: BR(config)# cdp holdtime ? <10-255> Length of time (in sec) that receiver must keep this packet

Abbreviating Commands

You have to enter only enough characters for the bridge to recognize the command as unique. This example shows how to enter the **show configuration** privileged EXEC command:

```
BR# show conf
```

Using no and default Forms of Commands

Most configuration commands also have 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 re-enable 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.

Understanding CLI Messages

Table 4-3 lists some error messages that you might encounter while using the CLI to configure your bridge.

Table 4-3 Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for your bridge to recognize the command.	Re-enter the command followed by a question mark (?) with a space between the command and the question mark. The possible keywords that you can enter with the command are displayed.
% Incomplete command.	You did not enter all the keywords or values required by this command.	Re-enter the command followed by a question mark (?) with a space between the command and the question mark. The possible keywords that you can enter with the command are displayed.
% 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 the commands that are available in this command mode. The possible keywords that you can enter with the command are displayed.

Using Command History

The IOS provides a history or record of commands that you have entered. This feature is particularly useful for recalling long or complex commands or entries, including access lists. You can customize the command history feature to suit your needs as described in these sections:

- [Changing the Command History Buffer Size, page 4-4](#)
- [Recalling Commands, page 4-5](#)
- [Disabling the Command History Feature, page 4-5](#)

Changing the Command History Buffer Size

By default, the bridge records ten command lines in its history buffer. Beginning in privileged EXEC mode, enter this command to change the number of command lines that the bridge records during the current terminal session:

```
BR# terminal history [size number-of-lines]
```

The range is from 0 to 256.

Beginning in line configuration mode, enter this command to configure the number of command lines the bridge records for all sessions on a particular line:

```
BR(config-line)# history [size number-of-lines]
```

The range is from 0 to 256.

Recalling Commands

To recall commands from the history buffer, perform one of the actions listed in [Table 4-4](#):

Table 4-4 Recalling Commands

Action ¹	Result
Press Ctrl-P or the up arrow key.	Recall commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Press Ctrl-N or the down arrow key.	Return 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.
show history	While in privileged EXEC mode, list the last several commands that you just entered. The number of commands that are displayed is determined by the setting of the terminal history global configuration command and history line configuration command.

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

Disabling the Command History Feature

The command history feature is automatically enabled.

To disable the feature during the current terminal session, enter the **terminal no history** privileged EXEC command.

To disable command history for the line, enter the **no history** line configuration command.

Using Editing Features

This section describes the editing features that can help you manipulate the command line. It contains these sections:

- [Enabling and Disabling Editing Features, page 4-6](#)
- [Editing Commands Through Keystrokes, page 4-6](#)
- [Editing Command Lines that Wrap, page 4-7](#)

Enabling and Disabling Editing Features

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

To re-enable the enhanced editing mode for the current terminal session, enter this command in privileged EXEC mode:

```
BR# terminal editing
```

To reconfigure a specific line to have enhanced editing mode, enter this command in line configuration mode:

```
BR(config-line)# editing
```

To globally disable enhanced editing mode, enter this command in line configuration mode:

```
BR(config-line)# no editing
```

Editing Commands Through Keystrokes

Table 4-5 shows the keystrokes that you need to edit command lines.

Table 4-5 Editing Commands Through Keystrokes

Capability	Keystroke ¹	Purpose
Move around the command line to make changes or corrections.	Ctrl-B or the left arrow key	Move the cursor back one character.
	Ctrl-F or the right arrow key	Move the cursor forward one character.
	Ctrl-A	Move the cursor to the beginning of the command line.
	Ctrl-E	Move the cursor to the end of the command line.
	Esc B	Move the cursor back one word.
	Esc F	Move the cursor forward one word.
	Ctrl-T	Transpose the character to the left of the cursor with the character located at the cursor.
Recall commands from the buffer and paste them in the command line. The bridge provides a buffer with the last ten items that you deleted.	Ctrl-Y	Recall the most recent entry in the buffer.
	Esc Y	Recall the next buffer entry. The buffer contains only the last 10 items that you have deleted or cut. If you press Esc Y more than ten times, you cycle to the first buffer entry.
Delete entries if you make a mistake or change your mind.	Delete or Backspace	Erase the character to the left of the cursor.
	Ctrl-D	Delete the character at the cursor.
	Ctrl-K	Delete all characters from the cursor to the end of the command line.
	Ctrl-U or Ctrl-X	Delete all characters from the cursor to the beginning of the command line.
	Ctrl-W	Delete the word to the left of the cursor.
	Esc D	Delete from the cursor to the end of the word.

Table 4-5 Editing Commands Through Keystrokes (continued)

Capability	Keystroke ¹	Purpose
Capitalize or lowercase words or capitalize a set of letters.	Esc C	Capitalize at the cursor.
	Esc L	Change the word at the cursor to lowercase.
	Esc U	Capitalize letters from the cursor to the end of the word.
Designate a particular keystroke as an executable command, perhaps as a shortcut.	Ctrl-V or Esc Q	
Scroll down a line or screen on displays that are longer than the terminal screen can display.	Return	Scroll down one line.
	Space	Scroll down one screen.
Note The <code>More</code> prompt appears for output that has more lines than can be displayed on the terminal screen, including <code>show</code> command output. You can use the Return and Space bar keystrokes whenever you see the <code>More</code> prompt.		
Redisplay the current command line if the bridge suddenly sends a message to your screen.	Ctrl-L or Ctrl-R	Redisplay the current command line.

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

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.

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.

In this example, the `access-list` global configuration command entry extends beyond one line. 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 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.

```
BR(config)# access-list 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1
BR(config)# $ 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1.20 255.25
BR(config)# $t tcp 131.108.2.5 255.255.255.0 131.108.1.20 255.255.255.0 eq
BR(config)# $108.2.5 255.255.255.0 131.108.1.20 255.255.255.0 eq 45
```

After you complete the entry, press **Ctrl-A** to check the complete syntax before pressing the **Return** key to execute the command. The dollar sign (\$) appears at the end of the line to show that the line has been scrolled to the right:

```
BR(config)# access-list 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1$
```

The software assumes you have a terminal screen that is 80 columns wide. If you have a width other than that, use the **terminal width** privileged EXEC command to set the width of your terminal.

Use line wrapping with the command history feature to recall and modify previous complex command entries. For information about recalling previous command entries, see the [“Editing Commands Through Keystrokes”](#) section on page 4-6.

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.

To use this functionality, enter a **show** or **more** command followed by the *pipe* character (|), one of the keywords **begin**, **include**, or **exclude**, and an expression that you want to search for or filter out:

```
command | {begin | include | exclude} regular-expression
```

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* are displayed.

This example shows how to include in the output display only lines where the expression *protocol* appears:

```
BR# show interfaces | include protocol
Vlan1 is up, line protocol is up
Vlan10 is up, line protocol is down
GigabitEthernet0/1 is up, line protocol is down
GigabitEthernet0/2 is up, line protocol is up
```

Accessing the CLI

You can open the bridge’s CLI using Telnet or Secure Shell (SSH).

Opening the CLI with Telnet

Follow these steps to open the CLI with Telnet. These steps are for a PC running Microsoft Windows with a Telnet terminal application. Check your PC operating instructions for detailed instructions for your operating system.

Step 1 Select **Start > Programs > Accessories > Telnet**.

If Telnet is not listed in your Accessories menu, select **Start > Run**, type **Telnet** in the entry field, and press **Enter**.

Step 2 When the Telnet window appears, click **Connect** and select **Remote System**.



Note In Windows 2000, the Telnet window does not contain drop-down menus. To start the Telnet session in Windows 2000, type **open** followed by the bridge's IP address.

- Step 3** In the Host Name field, type the bridge's IP address and click **Connect**.
- Step 4** At the username and password prompts, enter your administrator username and password. The default username is **Cisco**, and the default password is **Cisco**. The default enable password is also **Cisco**. Usernames and passwords are case-sensitive.
-

Opening the CLI with Secure Shell

Secure Shell Protocol is a protocol that provides a secure, remote connection to networking devices set up to use it. Secure Shell (SSH) is a software package that provides secure login sessions by encrypting the entire session. SSH features strong cryptographic authentication, strong encryption, and integrity protection. For detailed information on SSH, visit the homepage of SSH Communications Security, Ltd. at this URL: <http://www.ssh.com/>

SSH provides more security for remote connections than Telnet by providing strong encryption when a device is authenticated. See the “[Configuring the Bridge for Secure Shell](#)” section on page 5-16 for detailed instructions on setting up the bridge for SSH access.

