CHAPTER 2

Command Line Interface

This chapter describes the Cisco IOS Command Line Interface (CLI), how to use it to configure your switch module, and includes the following topics:

Understanding Command Modes

The Cisco IOS user interface is divided into many different modes. The commands available 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 switch module, 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 (shows the current configuration status), and **clear** commands (clears counters or interfaces). The user EXEC commands are not saved when the switch module 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 switch module 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 2-1 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 hostname *Switch*.

Table 2-1 Command Mode Summary

Mode	Access Method	Prompt	Exit Method	About This Mode
User EXEC	Start a session with the switch module	`Switch>`	Enter **logout** or **quit**	• Changes terminal settings • Performs basic tests • Displays system information
Privileged EXEC	While in user EXEC mode, enter the **enable** command	`Switch#`	Enter **disable** to exit	Verifies commands that you have entered. Use a password to protect access to this mode.
Global configuration	While in privileged EXEC mode, enter the configure command	`Switch(config)#`	To exit to privileged EXEC mode, enter **exit** or **end**, or press **Ctrl-Z**	Configures parameters that apply to the entire switch module
VLAN configuration	While in global configuration mode, enter the **vlan** *vlan-id* command.	`Switch(config-vlan)#`	To exit to global configuration mode, enter the **exit** command. To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**.	Configures VLAN parameters
Interface configuration	While in global configuration mode, enter the **interface** command (with a specific interface)	`Switch(config-if)#`	To exit to global configuration mode, enter **exit**. To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**.	Configures parameters for the Ethernet ports. For information about defining interfaces, see the "Using Interface Configuration Mode" section on page 8-13. To configure multiple interfaces with the same parameters, see the "Configuring a Range of Interfaces" section on page 8-15.
Line configuration	While in global configuration mode, specify a line with the **line vty** or **line console** command	`Switch(config-line)#`	To exit to global configuration mode, enter **exit**. To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**.	Configures parameters for the terminal line

Understanding the CLI Help System

To display a list of commands available for each command mode, enter a question mark (?) at the system prompt. You can also obtain a list of associated keywords and arguments for any command, as shown in Table 2-2.

Table 2-2 CLI Help Summary

Command	Description
help	Brief description of the help system in any command mode

*abbreviated-command-entry***?**	List of commands that begin with a particular character string. For example:
	`Switch# di?`
	`dir disable disconnect`

abbreviated-command-entry<**Tab**>	Complete a partial command name. For example:
	`Switch# sh conf<tab>`
	`Switch# show configuration`

?	List all commands available for a particular command mode. For example: `Switch> ?`
command **?**	List the associated keywords for a command. For example: `Switch> show ?`

command keyword **?**	List the associated arguments for a keyword. For example:
	`Switch(config)# cdp holdtime ?`
	`  <10-255> Length of time (in sec) that receiver must keep this packet`

Understanding Abbreviated Commands

You need to enter only enough characters for the switch module to recognize the command as unique. This example shows how to enter the **show configuration** privileged EXEC command in an abbreviated form:

```
Switch# show conf
```

Understanding *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 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 Error Messages

Table 2-3 lists some error messages that you might encounter while using the CLI to configure your switch module.

Table 2-3 Common CLI Error Messages

Error Message	Meaning	How to Get Help
`% Ambiguous command: "show con"`	You did not enter enough characters for your switch module 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 display.
`% Incomplete command`	You did not enter all the keywords or values required by this command	
`% Invalid input detected at ‘^’ marker`	You entered the command incorrectly - he caret (^) marks the point of the error	

Using 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 as described in these sections:

- Changing the Command History Buffer Size, page 2-4 (optional)
- Recalling Commands, page 2-5 (optional)
- Disabling the Command History Feature, page 2-5 (optional)

Changing the Command History Buffer Size

By default, the switch module 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. These procedures are optional.

Beginning in privileged EXEC mode, enter this command to change the number of command lines that the switch module records during the current terminal session (range is from 0 to 256):

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

Beginning in line configuration mode, enter this command to configure the number of command lines the switch module records for all sessions on a particular line (range is from 0 to 256):

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

Recalling Commands

To recall commands from the history buffer, perform one of the optional actions listed in Table 2-4.

Table 2-4 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 appear is controlled by the setting of the **terminal history** global configuration command and the **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. You can disable it for the current terminal session or for the command line. These procedures are optional.

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.

- Enabling and Disabling Editing Features, page 2-6 (optional)
- Editing Commands Through Keystrokes, page 2-6 (optional)
- Editing Command Lines That Wrap, page 2-8 (optional)

Enabling and Disabling Editing Features

Although enhanced editing mode is automatically enabled, you can disable it, re-enable it, or configure a specific line to have enhanced editing. These procedures are optional.

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

```
The arrow keys function only on ANSI-compatible terminals such as VT100s.Switch
(config-line)# no editing
```

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

```
Switch# terminal editing
```

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

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

Editing Commands Through Keystrokes

Table 2-5 shows the keystrokes that you need to edit command lines. These keystrokes are optional.

Table 2-5 Editing Commands through Keystrokes

Capability	Keystroke[1]	Description
Move around the command line to make changes or corrections	Press **Ctrl-B**, or press the left arrow key	Move the cursor back one character
	Press **Ctrl-F**, or press the right arrow key	Move the cursor forward one character
	Press **Ctrl-A**	Move the cursor to the beginning of the command line
	Press **Ctrl-E**	Move the cursor to the end of the command line
	Press **Esc B**	Move the cursor back one word
	Press **Esc F**	Move the cursor forward one word
	Press **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 switch module provides a buffer with the last ten items that you deleted.	Press **Ctrl-Y**	Recall the most recent entry in the buffer

Table 2-5 Editing Commands through Keystrokes (continued)

Capability	Keystroke[1]	Description
	Press **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	Press **Delete** or **Backspace**	Erase the character to the left of the cursor
	Press **Ctrl-D**	Delete the character at the cursor
	Press **Ctrl-K**	Delete all characters from the cursor to the end of the command line
	Press **Ctrl-U** or **Ctrl-X**	Delete all characters from the cursor to the beginning of the command line
	Press **Ctrl-W**	Delete the word to the left of the cursor
	Press **Esc D**	Delete from the cursor to the end of the word
Capitalize or lowercase words or capitalize a set of letters	Press **Esc C**	Capitalize at the cursor
	Press **Esc L**	Change the word at the cursor to lowercase
	Press **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	Press **Ctrl-V** or **Esc Q**	
Scroll down a line or screen on displays that are longer than the terminal screen can display. **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.	Press **Return**	Scroll down one line

Table 2-5 Editing Commands through Keystrokes (continued)

Capability	Keystroke[1]	Description
	Press the **Space** bar	Scroll down one screen
Redisplay the current command line if the switch module suddenly sends a message to your screen.	Press **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. 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.

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.

```
Switch(config)# access-list 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1
Switch(config)# $ 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1.20 255.25
Switch(config)# $t tcp 131.108.2.5 255.255.255.0 131.108.1.20 255.255.255.0 eq
Switch(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:

```
Switch(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 2-6.

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. Using these commands is optional.

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* appear.

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

```
Switch# 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
```