Using the Command-Line Interface (CLI)

The Cisco IOS command-line interface (CLI) is the primary user interface used for configuring, monitoring, and maintaining Cisco devices. This user interface allows you to directly and simply execute Cisco IOS commands, whether using a router console or terminal, or using remote access methods.

This chapter describes the basic features of the Cisco IOS command-line interface and how to use them. Topics covered include navigation and editing features, help features, command history features, and Cisco IOS command modes.

Additional user interfaces include setup mode (used for first time startup), the Cisco Web Browser, and user menus configured by a system administrator. For information about setup mode, see the "Using Configuration Tools" chapter of this book. For information on issuing commands using the Cisco Web Browser, see the "Using the Cisco Web Browser" chapter of this book. For information on user-menus, see the "Managing Connections, Menus, and System Banners" chapter of this book.

For a complete description of the user interface commands in this chapter, refer to the "Basic Command-Line Interface Commands" chapter of the *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands, use the command reference index or search online.

This chapter contains information about the following command-line interface topics:

- Understanding Cisco IOS Command Modes
- Getting Context-Sensitive Help Within a Command Mode
- Checking Command Syntax
- Using CLI Command History
- Using the No and Default Forms of Commands
- Using Command-Line Editing Features and Shortcuts
- Searching and Filtering CLI Output

Understanding Cisco IOS Command Modes

The Cisco IOS Command-Line Interface is divided into many different command modes. Each command mode has its own set of commands available for the configuration, maintenance, and monitoring of router and network operations. The commands available to you at any given time depend on which mode you are currently in. Entering a question mark (**?**) at the system prompt (router prompt) allows you to obtain a list of commands available for each command mode.

The use of specific commands allows you to navigate from one command mode to another. The basic hierarchy of these command modes is as follows:
user EXEC mode->privileged EXEC mode->global configuration mode->
specific configuration modes->configuration submodes->configuration subsubmodes.

When you start a session on the router, you begin in *user EXEC mode*. For security purposes, only a limited subset of EXEC commands are available in user EXEC mode. This level of access is reserved for tasks which do not change the configuration of the router, like checking the router status.

In order to have access to all commands, you must enter *privileged EXEC mode*. Normally, you must enter a password to enter privileged EXEC mode. From privileged EXEC mode, you can enter any EXEC command. Most of the 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 EXEC commands are not saved across reboots of the router.

From privileged EXEC mode, you can enter *global configuration mode*. In global configuration mode, you can enter commands which configure general system characteristics. Global configuration mode is also used to enter specific configuration modes. Configuration modes allow you to make changes to the running configuration. If you later save the configuration, these commands are stored across router reboots.

From global configuration mode you can enter a variety of protocol-specific or feature-specific configuration modes. The CLI hierarchy requires that you enter these specific configuration modes only through global configuration mode. As an example, this chapter describes *interface configuration mode*, a commonly used configuration mode.

From configuration modes, you can enter configuration submodes. Configuration submodes are used for the configuration of specific features within the scope of the configuration mode. As an example, this chapter describes the *subinterface configuration submode*.

ROM monitor mode is a separate mode used when the router cannot boot properly. If your router or access server does not find a valid system image when it is booting, or if its configuration file is corrupted at startup, the system may enter ROM monitor mode. You may also enter ROM monitor mode by using the Break key to interrupt system startup.

The above information is described in more detail in the following sections:

- User EXEC Mode
- Privileged EXEC Mode
- Global Configuration Mode
- Interface Configuration Mode
- Subinterface Configuration Mode
- ROM Monitor Mode

These sections are followed by a table (Table 1) which summarizes these command modes.

User EXEC Mode

After you log in to the router or access server, you are automatically in user EXEC command mode. The EXEC commands available at the user level are a subset of those available at the privileged level. In general, the user EXEC commands allow you to connect to remote devices, change terminal settings on a temporary basis, perform basic tests, and list system information.

To list the user EXEC commands, use the following command:

Command	Purpose
?	Lists the user EXEC commands.

The user-level prompt consists of the host name followed by the angle bracket (>):

```
Router>
```

The default host name is Router unless it has been changed during initial configuration using the **setup** command. Refer to the product user guide for information on the **setup** facility. You can also change the host name using the **hostname** global configuration command described in the "Basic System Management Commands" chapter in the *Cisco IOS Configuration Fundamentals Command Reference*.

To list the commands available in user EXEC mode, enter a question mark (**?**) as shown in the following example:

```
Router> ?
Exec commands:
 <1-99>          Session number to resume
 connect         Open a terminal connection
 disconnect      Disconnect an existing telnet session
 enable          Turn on privileged commands
 exit            Exit from the EXEC
 help            Description of the interactive help system
 lat             Open a lat connection
 lock            Lock the terminal
 login           Log in as a particular user
 logout          Exit from the EXEC
 menu            Start a menu-based user interface
 mbranch         Trace multicast route for branch of tree
 mrbranch        Trace reverse multicast route to branch of tree
 mtrace          Trace multicast route to group
 name-connection Name an existing telnet connection
 pad             Open a X.29 PAD connection
 ping            Send echo messages
 resume          Resume an active telnet connection
 show            Show running system information
 systat          Display information about terminal lines
 telnet          Open a telnet connection
 terminal        Set terminal line parameters
 tn3270          Open a tn3270 connection
 trace           Trace route to destination
 where           List active telnet connections
 x3              Set X.3 parameters on PAD
 xremote         Enter XRemote mode
```

The list of commands will vary depending on the software feature set and which router platform you are using.

> **Note** You can enter commands in uppercase, lowercase, or mixed case. Only passwords are case sensitive. However, it is a Cisco IOS documentation convention to always present commands in lowercase.

Privileged EXEC Mode

Because many of the privileged commands set operating parameters, privileged access should be password protected to prevent unauthorized use. The privileged command set includes those commands contained in user EXEC mode, as well as the **configure** command through which you can access the remaining command modes. Privileged EXEC mode also includes high-level testing commands, such as **debug**. For details on the **debug** commands, see the *Cisco IOS Debug Command Reference*.

The privileged EXEC mode prompt consists of the devices's host name followed by the pound sign (#), as shown in the following example:

```
Router#
```

> **Note** Examples in Cisco IOS documentation assume the use of the default name of "Router". Different devices (for example, access servers) may use a different default name. If the router or access server has been named with the **hostname** command, that name will appear as the prompt instead of the default name.

To access and list the privileged EXEC commands, use the following commands:

	Command	Purpose
Step 1	Router> **enable** [*password*]	Enters the privileged EXEC mode.
Step 2	Router# **?**	Lists privileged EXEC commands.

If the system administrator has set a password, you are prompted to enter it before being allowed access to privileged EXEC mode. The password is not displayed on the screen and is case sensitive. If an enable password has not been set, enable mode can be accessed only from the router console. The system administrator uses the **enable password** global configuration command to set the password that restricts access to privileged mode. This command is described in the "Passwords and Privileges Commands" chapter in the *Cisco IOS Security Command Reference*.

To return to user EXEC mode, use the following command:

Command	Purpose
Router# **disable**	Returns you to user EXEC mode from privileged EXEC mode.

The following example shows how to access privileged EXEC mode:

```
Router> enable
Password:letmein
Router#
```

Note that the password will not be displayed as you type, but is shown here for illustrational purposes. From the privileged level, you can access global configuration mode, as described in the following section.

Global Configuration Mode

The term "global" is used to indicate characteristics or features that affect the system as a whole. Global configuration mode is used to configure your system globally, or to enter specific configuration modes to configure specific elements such as interfaces or protocols. Use the **configure terminal** privileged EXEC command to enter global configuration mode.

To access global configuration mode, use the following command in privileged EXEC mode:

Command	Purpose
Router# **configure terminal**	From privileged EXEC mode, enters global configuration mode.

The following example shows the process of entering global configuration mode from privileged EXEC mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Note that the system prompt changes to indicate that you are now in global configuration mode. The prompt for global configuration mode consists of the host-name of the device followed by (config) and the pound sign (#). To list the commands available in privileged EXEC mode, issue the **?** command at the prompt.

Commands entered in global configuration mode update the running configuration file as soon as they are entered. In other words, changes to the configuration take effect each time you press the Enter or Return key at the end of a valid command. However, these changes are not saved into the startup configuration file until you issue the **copy running-config startup-config** EXEC mode command. This behavior is explained in more detail later in this document.

As shown in the example above, the system dialogue prompts you to end your configuration session (exit configuration mode) by pressing the Control (Ctrl) and "z" keys simultaneously; when you press these keys, **^Z** is printed to the screen. You can actually end your configuration session by entering the Ctrl-Z key combination, using the **end** command, using the Ctrl-C key combination. The **end** command is the recommended way to indicate to the system that you are done with the current configuration session.

Warning **If you use Ctrl-Z at the end of a command line in which a valid command has been typed, that command will be added to the running configuration file. In other words, using Ctrl-Z is equivalent to hitting the Enter (Carriage Return) key before exiting. For this reason, it is safer to end your configuration session using the end command. Alternatively, you can use the Ctrl-C key combination to end your configuration session without sending a Carriage Return signal.**

You can also use the **exit** command to return from global configuration mode to EXEC mode, but this only works in global configuration mode. Pressing Ctrl-Z or entering the **end** command will always take you back to EXEC mode regardless of which configuration mode or configuration submode you are in.

To exit global configuration command mode and return to privileged EXEC mode, use one of the following commands:

Command	Purpose

`Router(config)# end`
or
 	
`Router(config)# ^Z`
	Ends the current configuration session and returns to privileged EXEC mode.

`Router(config)# exit`	Exits the current command mode and returns to the preceding mode. For example, exits from global configuration mode to privileged EXEC mode.

From global configuration mode, you can enter a number of protocol-specific, platform-specific, and feature-specific configuration modes. For a complete list of configuration modes, see the "Cisco IOS Command Modes" appendix in this book. This appendix provides references to the appropriate documentation module for information about specific configuration modes.

Interface configuration mode, described in the following section, is an example of a configuration mode you can enter from global configuration mode.

Interface Configuration Mode

One example of a specific configuration mode you enter from global configuration mode is interface configuration mode.

Many features are enabled on a per-interface basis. Interface configuration commands modify the operation of an interface such as an Ethernet, FDDI, or serial port. Interface configuration commands always follow an **interface** global configuration command, which defines the interface type.

For details on interface configuration commands that affect general interface parameters, such as bandwidth, clock rate, and so on, see the *Cisco IOS Interface Command Reference*. For protocol-specific commands, see the appropriate Cisco IOS software command reference.

To access and list the interface configuration commands, use the following commands:

	Command	Purpose
Step 1	`Router(config)# interface type number`	From global configuration mode, enters interface configuration mode.
Step 2	`Router(config-if)# ?`	Lists the interface configuration commands.

In the following example, serial interface 0 is about to be configured. The new prompt `Router(config-if)#` indicates interface configuration mode.

```
Router(config)# interface serial 0 <Return>
Router(config-if)#
```

To exit interface configuration mode and return to global configuration mode, enter the **exit** command.

Configuration submodes are configuration modes reached from other configuration modes (besides global configuration mode). Configuration submodes are for the configuration of specific elements in the configuration mode. For a complete list of configuration submodes, see the "Cisco IOS Command Modes" appendix in this book. One example of a configuration submode is subinterface configuration submode, described in the following section.

Subinterface Configuration Submode

From interface configuration mode, you can enter subinterface configuration submode. In this submode you can configure multiple virtual interfaces (called subinterfaces) on a single physical interface. Subinterfaces appear to be distinct physical interfaces to the various protocols. For example, Frame Relay networks provide multiple point-to-point links called permanent virtual circuits (PVCs). PVCs can be grouped under separate subinterfaces that in turn are configured on a single physical interface. From a bridging spanning-tree viewpoint, each subinterface is a separate bridge port, and a frame arriving on one subinterface can be sent out on a another subinterface.

Subinterfaces also allow multiple encapsulations for a protocol on a single interface. For example, a router or access server can receive an ARPA-framed IPX packet and forward the packet back out the same physical interface as a SNAP-framed IPX packet.

For detailed information on how to configure subinterfaces, see the appropriate module for a specific protocol in the Cisco IOS software documentation.

To access and list the subinterface configuration commands, use the following commands:

	Command	Purpose
Step 1	See the example that follows. For information on interface commands that allow subinterface implementation, see the protocol specific chapter later in this publication.	From interface configuration mode, configures a virtual interface.
Step 2	?	Lists the subinterface configuration commands.

In the following example, a subinterface is configured for serial line 2, which is configured for Frame Relay encapsulation. The subinterface is called 2.1 to indicate that it is subinterface 1 of serial interface 2. The new prompt `Router(config-subif)#` indicates that you are in subinterface configuration mode. The subinterface can be configured to support one or more Frame Relay PVCs.

```
Router(config)# interface serial 2
Router(config-if)# encapsulation frame-relay
Router(config-if)# interface serial 2.1
Router(config-subif)#
```

To exit subinterface configuration mode and return to global configuration mode, enter the **exit** command. To exit configuration mode and return to privileged EXEC mode, press **Ctrl-Z**.

ROM Monitor Mode

If your router or access server does not find a valid system image to load, the system will enter read-only memory (ROM) monitor mode. ROM monitor (ROMMON) mode can also be accessed by interrupting the boot sequence during startup. From ROM monitor mode, you can boot the device or perform diagnostic tests.

On most routers or access servers you can enter ROM monitor mode by entering the **reload** EXEC command and then pressing the Break key during the first 60 seconds of startup (the default ASCII configuration for Break is Ctrl-C).

To access and list the ROM monitor configuration commands, use the following commands:

	Command	Purpose
Step 1	`reload`	Begins reloading of system software image.
Step 2	Press the Break key (default is Ctrl-C) during the first 60 seconds while the system is booting.	Interrupts the boot sequence and enters ROM monitor mode from privileged EXEC mode.
Step 3	`?`	Lists the ROM monitor commands.

The ROM monitor mode is indicated by the angle bracket (>) prompt. On some Cisco routers the default ROM monitor prompt is rommon >. A list of ROM monitor commands are displayed when you enter the **?** command or **help** command. The following example shows how this list of commands may appear:

```
User break detected at location 0x8162ac6\Œ
rommon 1 > ?
alias             set and display aliases command
boot              boot up an external process
break             set/show/clear the breakpoint
confreg           configuration register utility
cont              continue executing a downloaded image
context           display the context of a loaded image
cpu_card_type     display CPU card type
dev               list the device table
dir               list files in file system
dis               disassemble instruction stream
frame             print out a selected stack frame
help              monitor builtin command help
history           monitor command history
meminfo           main memory information
repeat            repeat a monitor command
reset             system reset
set               show all monitor variables
stack             produce a stack trace
sync              write monitor environment to NVRAM
sysret            print out info from last system return
unalias           unset an alias
unset             unset a monitor variable
rommon 2>
```

The list of available commands will depend on the software image and platform you are using. Some versions of ROMMON will display a list of commands in an pre-aliased format such as the following:

```
> ?
$ state       Toggle cache state (? for help)
B [filename] [TFTP Server IP address | TFTP Server Name]
              Load and execute system image from ROM or from TFTP server
C [address]  Continue execution [optional address]
D /S M L V    Deposit value V of size S into location L with modifier M
E /S M L      Examine location L with size S with modifier M
G [address]  Begin execution
H             Help for commands
I             Initialize
K             Stack trace
L [filename] [TFTP Server IP address | TFTP Server Name]
              Load system image from ROM or from TFTP server, but do not
              begin execution
O             Show configuration register option settings
P             Set the break point
S             Single step next instruction
T function   Test device (? for help)
Deposit and Examine sizes may be B (byte), L (long) or S (short).
Modifiers may be R (register) or S (byte swap).
Register names are: D0-D7, A0-A6, SS, US, SR, and PC
```

To exit ROM Monitor mode, use the **continue** command or **C** command alias. If you have changed the configuration, use the **copy running-config startup-config** command to save your configuration changes, then issue the **reload** command.

For more information on ROM monitor mode characteristics (including using aliases for commands) and using ROM monitor mode, see the "Rebooting a Router" chapter in this document.

Summary of Main Cisco IOS Command Modes

Table 1 summarizes the main command modes used in the Cisco IOS CLI. For a complete list of configuration modes, see the "Cisco IOS Command Modes" appendix in this book.

Table 1   Summary of the Base Cisco IOS Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	Router>	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** EXEC command.	Router#	To exit back to user EXEC mode, use the **disable** command. To enter global configuration mode, use the **configure terminal** privileged EXEC command.
Global configuration	From privileged EXEC mode, use the **configure terminal** privileged EXEC command.	Router(config)#	To exit to privileged EXEC mode, use the **exit** or **end** command or press **Ctrl-Z**. To enter interface configuration mode, use the **interface** configuration command.

Table 1   Summary of the Base Cisco IOS Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method
Interface configuration	From global configuration mode, enter by specifying an interface with an **interface** command.	`Router(config-if)#`	To exit to global configuration mode, use the **exit** command. To exit to privileged EXEC mode, use the **exit** command or press **Ctrl-Z**. To enter subinterface configuration mode, specify a subinterface with the **interface** command.
Subinterface configuration	From interface configuration mode, specify a subinterface with an **interface** command.	`Router(config-subif)#`	To exit to global configuration mode, use the **exit** command. To enter privileged EXEC mode, use the **end** command or press **Ctrl-Z**.

ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the Break key during the first 60 seconds while the system is booting.	`>`
		
		or
		
		`rommon >`
			If you entered ROM monitor mode by interrupting the loading process, you can exit ROM monitor and resume loading by using the **continue** command or **C** command alias.

Getting Context-Sensitive Help Within a Command Mode

Entering a question mark (**?**) at the system prompt displays a list of commands available for each command mode. You can also get a list of any command's associated keywords and arguments with the context-sensitive help feature.

To get help specific to a command mode, a command, a keyword, or an argument, perform one of the following commands:

Command	Purpose
`help`	Obtain a brief description of the help system in any command mode.
`abbreviated-command-entry`**?**	Obtain a list of commands that begin with a particular character string.
`abbreviated-command-entry`**<Tab>**	Complete a partial command name.
?	List all commands available for a particular command mode.
`command` **?**	List a command's associated keywords.
`command keyword` **?**	List a keyword's associated arguments.

When using context-sensitive help, the space (or lack of a space) before the question mark (**?**) is significant. To obtain a list of commands that begin with a particular character sequence, type in those characters followed immediately by the question mark (**?**). Do not include a space. This form of help is called *word help,* because it completes a word for you.

To list keywords or arguments, enter a question mark (**?**) in place of a keyword or argument. Include a space before the **?**. This form of help is called *command syntax help,* because it reminds you which keywords or arguments are applicable based on the command, keywords, and arguments you already have entered.

You can abbreviate commands and keywords to the number of characters that allow a unique abbreviation. For example, you can abbreviate the **configure terminal** command to **config term**. Because the shortened form of the command is unique, the router will accept the shorted form and execute the command.

Enter the **help** command (which is available in any command mode) for a brief description of the help system:

```
Router# help
Help may be requested at any point in a command by entering
a question mark '?'. If nothing matches, the help list will
be empty and you must back up until entering a '?' shows the
available options.
Two styles of help are provided:
1. Full help is available when you are ready to enter a
   command argument (e.g. 'show ?') and describes each possible
   argument.
2. Partial help is provided when an abbreviated argument is entered
   and you want to know what arguments match the input
   (e.g. 'show pr?'.)
```

As described in the **help** command output, you can enter a partial command name and a question mark (**?**) to obtain a list of commands beginning with a particular character set. (See the section "Completing a Partial Command Name" later in this chapter for more details.)

Example of Context Sensitive Help

The following example illustrates how the context-sensitive help feature enables you to create an access list from configuration mode.

Enter the letters **co** at the system prompt followed by a question mark (**?**). Do not leave a space between the last letter and the question mark (**?**). The system provides the commands that begin with **co**.

```
Router# co?
configure   connect   copy
```

Enter the **configure** command followed by a space and a question mark (**?**) to list the command's keywords and a brief explanation:

```
Router# configure ?
  memory    Configure from NV memory
  network   Configure from a TFTP network host
  overwrite-network  Overwrite NV memory from TFTP network host
  terminal  Configure from the terminal
  <cr>
```

The <cr> symbol (Carriage Return) appears in the list to indicate that one of your options is to press the Return (or Enter) key to execute the command, without adding any additional keywords. In the example above, if you enter the **configure** command followed by the Carriage Return (Enter or Return key), you will be prompted to specify terminal, memory, or network.

Enter the **terminal** keyword to enter configuration mode from the terminal:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Enter the **access-list** command followed by a space and a question mark (**?**) to list the command's keywords:

```
Router(config)# access-list ?
  <1-99>      IP standard access list
  <100-199>   IP extended access list
  <1000-1099> IPX SAP access list
  <1100-1199> Extended 48-bit MAC address access list
  <200-299>   Protocol type-code access list
  <300-399>   DECnet access list
  <400-499>   XNS standard access list
  <500-599>   XNS extended access list
  <600-699>   Appletalk access list
  <700-799>   48-bit MAC address access list
  <800-899>   IPX standard access list
  <900-999>   IPX extended access list
```

The two numbers within the angle brackets represent an inclusive range. Enter the access list number **99** and then enter another question mark (**?**) to see the arguments that apply to the keyword and brief explanations:

```
Router(config)# access-list 99 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
```

Enter the **deny** argument followed by a question mark (**?**) to list additional options:

```
Router(config)# access-list 99 deny ?
  A.B.C.D  Address to match
```

Generally, uppercase letters represent variables, though this is not always the case. Enter the IP address followed by a question mark (**?**) to list additional options:

```
Router(config)# access-list 99 deny 131.108.134.0 ?
  A.B.C.D  Mask of bits to ignore
  <cr>
```

In the above example, the variables A.B.C.D. indicate that use of a wildcard mask is allowed. The wildcard mask is a method for matching IP addresses or ranges of IP addresses. For example, a wildcard mask of 0.0.0.255 matches any number in the range from 0 to 255 that appears in the fourth octet of an IP address.

Enter the wildcard mask followed by a question mark (**?**) to list further options.

```
Router(config)# access-list 99 deny 131.108.134.0 0.0.0.255 ?
<cr>
```

The <cr> symbol by itself indicates there are no more keywords or arguments. Press Return to execute the command.

```
Router(config)# access-list 99 deny 131.108.134.0 0.0.0.255
```

The system adds an entry to access list 99 that denies access to all hosts on subnet 131.108.134.0, while ignoring bits for IP addresses that end in 0 to 255.

Display Help for All User-Level Commands

To configure a line to display help for the full set of user-level commands during all sessions, use the following commands in line configuration mode:

Command	Purpose
`full-help`	Configure a line or lines to receive help for the full set of user-level commands when a user presses **?**.

To configure the current session to display help for the full set of user-level commands, use the following command in user or privileged EXEC mode:

Command	Purpose
`terminal full-help`	Configure this session to provide help for the full set of user-level commands.

The **full-help** and **terminal full-help** commands enable (or disable) a display of all help messages available from the terminal. They are used with the **show** command.

The following example is output for **show ?** with **terminal full-help** disabled and then enabled:

```
Router> show ?
  bootflash  Boot Flash information
  calendar   Display the hardware calendar
  clock      Display the system clock
  context    Show context information
  dialer     Dialer parameters and statistics
  history    Display the session command history
  hosts      IP domain-name, lookup style, nameservers, and host table
  isdn       ISDN information
  kerberos   Show Kerberos Values
  modemcap   Show Modem Capabilities database
  ppp        PPP parameters and statistics
  rmon       rmon statistics
  sessions   Information about Telnet connections
  snmp       snmp statistics
  terminal   Display terminal configuration parameters
  users      Display information about terminal lines
  version    System hardware and software status

Router> terminal full-help
Router> show ?
  access-expression  List access expression
  access-lists       List access lists
  aliases            Display alias commands
  apollo             Apollo network information
  appletalk          AppleTalk information
  arp                ARP table
  async              Information on terminal lines used as router interfaces
  bootflash          Boot Flash information
  bridge             Bridge Forwarding/Filtering Database [verbose]
  bsc                BSC interface information
```

```
bstun             BSTUN interface information
buffers           Buffer pool statistics
calendar          Display the hardware calendar
cdp               CDP information
clns              CLNS network information
clock             Display the system clock
cls               DLC user information
cmns              Connection-Mode networking services (CMNS) information
...
x25               X.25 information
xns               XNS information
xremote           XRemote statistics
```

Checking Command Syntax

The user interface provides error isolation in the form of an error indicator, a caret symbol (^). The ^ symbol appears at the point in the command string where you have entered an incorrect command, keyword, or argument.

In the following example, suppose you want to set the clock. Use context-sensitive help to check the syntax for setting the clock.

```
Router# clock ?
  set  Set the time and date
Router# clock
```

The help output shows that the **set** keyword is required. Check the syntax for entering the time:

```
Router# clock set ?
hh:mm:ss   Current time
Router# clock set
```

Enter the current time:

```
Router# clock set 13:32:00
% Incomplete command.
```

The system indicates that you need to provide additional arguments to complete the command. Press **Ctrl-P** (see the next section, "Use the Command History Features") to automatically repeat the previous command entry. Then add a space and question mark (**?**) to reveal the additional arguments:

```
Router# clock set 13:32:00 ?
  <1-31>     Day of the month
  January    Month of the year
  February
  March
  April
  May
  June
  July
  August
  September
  October
  November
  December
```

Now you can complete the command entry:

```
Router# clock set 13:32:00 23 February 97
                                   ^
% Invalid input detected at '^' marker.
```

The caret symbol (^) and help response indicate an error at 97. To list the correct syntax, enter the command up to the point where the error occurred and then enter a question mark (**?**):

```
Router# clock set 13:32:00 23 February ?
  <1993-2035> Year
Router# clock set 13:32:00 23 February
```

Enter the year using the correct syntax and press Return to execute the command.

```
Router# clock set 13:32:00 23 February 1997
```

Using CLI Command History

The Cisco IOS CLI 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. With the command history feature, you can complete the tasks in the following sections:

- Setting the Command History Buffer Size
- Recalling Commands
- Disabling the Command History Feature

Setting the Command History Buffer Size

By default, the system records 10 command lines in its history buffer. To set the number of command lines that the system will record during the current terminal session, use the following command in EXEC mode:

Command	Purpose
`terminal history` [`size` *number-of-lines*]	Enable the command history feature for the current terminal session.

The **terminal no history size** command resets the number of lines saved in the history buffer to the default of 10 lines.

To configure the number of command lines the system will record for all sessions on a particular line, use the following command in line configuration mode:

Command	Purpose
`history` [`size` *number-of-lines*][1]	Enable the command history feature.

1. The **no history** command turns off command history for the line.

Recalling Commands

To recall commands from the history buffer, use one of the following commands:

Command	Purpose
Press **Ctrl-P** or the up arrow key.[1]	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.[1]	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 EXEC mode, list the last several commands you have just entered.

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 it during the current terminal session, use the following EXEC mode command:

Command	Purpose
terminal no history	Disable the command history feature for the current session.

To configure a specific line so that the command history feature is disabled, use the following command in line configuration mode:

Command	Purpose
no history	Configure the line so that the command history feature is disabled.

Using the 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. Use the command without the keyword **no** to reenable a disabled feature or to enable a feature that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** form of the **ip routing** command. To reenable it, use the plain **ip routing** form. The Cisco IOS software command reference publications provide the complete syntax for every configuration command and describes what the **no** form of a command does (when a **no** form is available).

Using Command-Line Editing Features and Shortcuts

There are a variety of shortcuts and editing features enabled for the Cisco IOS command-line interface. The following subsections describe these features:

- Moving Around on the Command Line
- Completing a Partial Command Name
- Pasting in Buffer Entries
- Editing Command Lines that Wrap
- Deleting Entries
- Scrolling Down a Line or a Screen
- Redisplaying the Current Command Line
- Transposing Mistyped Characters
- Controlling Capitalization
- Designating a Keystroke as a Command Entry
- Disabling and Reenabling Enhanced Editing Features

Moving Around on the Command Line

Use the following commands to move the cursor around on the command line to make corrections or changes:

	Keystrokes	Purpose
Step 1	Press **Ctrl-B** or press the left arrow key.[1]	Move the cursor back one character.
Step 2	Press **Ctrl-F** or press the right arrow key.[1]	Move the cursor forward one character.
Step 3	Press **Ctrl-A**.	Move the cursor to the beginning of the command line.
Step 4	Press **Ctrl-E**.	Move the cursor to the end of the command line.
Step 5	Press **Esc B**.	Move the cursor back one word.
Step 6	Press **Esc F**.	Move the cursor forward one word.

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

Completing a Partial Command Name

If you cannot remember a complete command name, press the Tab key to allow the system to complete a partial entry. To do so, use the following command:

Keystrokes	Purpose
Enter the first few letters and press the Tab key.	Complete a command name.

If your keyboard does not have a Tab key, press **Ctrl-I** instead.

In the following example, when you enter the letters **conf** and press the Tab key, the system provides the complete command:

```
Router# conf<Tab>
Router# configure
```

Instead of immediately executing the command, the CLI displays the full command name and waits for you to use the carriage return (Return or Enter) key. This way you can 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, the system beeps to indicate an error.

If there is an error, enter a question mark (**?**) to obtain a list of commands that begin with that set of characters. Do not leave a space between the last letter you enter and the question mark (**?**).

For example, there are three commands in privileged mode that start with co. To see what they are, type **co?** at the privileged EXEC prompt:

```
Router# co?
configure  connect  copy
Router# co
```

Pasting in Buffer Entries

The system provides a buffer that contains the last 10 items you deleted. To recall these items and paste them in the command line, use the following commands:

	Keystrokes	Purpose
Step 1	Press **Ctrl-Y**.	Recall the most recent entry in the buffer.
Step 2	Press **Esc Y**.	Recall the next buffer entry.

The buffer contains only the last 10 items you have deleted or cut. If you press **Esc Y** more than 10 times, you will cycle back to the first buffer entry.

Editing Command Lines that Wrap

The enhanced editing provides 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 10 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, use the following command:

Keystrokes	Purpose
Press **Ctrl-B** or the left arrow key repeatedly until you scroll back to the beginning of the command entry, or press **Ctrl-A** to return directly to the beginning of the line.[1]	Return to the beginning of a command line to verify that you have entered a lengthy command correctly.

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

In the following example, the **access-list** command entry extends beyond one line. When the cursor first reaches the end of the line, the line is shifted 10 spaces to the left and redisplayed. The dollar sign ($) indicates that the line has been scrolled to the left. Each time the cursor reaches the end of the line, the line is again shifted 10 spaces to the left.

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

When you have completed 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 indicate that the line has been scrolled to the right:

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

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

Use line wrapping in conjunction with the command history feature to recall and modify previous complex command entries. See the section "Recall Commands" in this chapter for information about recalling previous command entries.

Deleting Entries

Use any of the following commands to delete command entries if you make a mistake or change your mind:

Keystrokes	Purpose
Press the Delete or Backspace key.	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.

Scrolling Down a Line or a Screen

When you use the help facility to list the commands available in a particular mode, the list is often longer than the terminal screen can display. In such cases, a More prompt is displayed at the bottom of the screen, assuming that the **length** or **terminal length** command is configured correctly. To view the next line or screen, use the following commands:

Keystrokes	Purpose
Press the Return key.	Scroll down one line.
Press the Space bar.	Scroll down one screen.

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 keystrokes listed above whenever you see the More prompt.

Redisplaying the Current Command Line

If you are entering a command and the system suddenly sends a message to your screen, you can easily recall your current command line entry. To do so, use the following command:

Keystrokes	Purpose
Press **Ctrl-L** or **Ctrl-R**.	Redisplay the current command line.

Transposing Mistyped Characters

If you have mistyped a command entry, you can transpose the mistyped characters by using the following command:

Keystrokes	Purpose
Press **Ctrl-T**.	Transpose the character to the left of the cursor with the character located at the cursor.

Controlling Capitalization

You can capitalize or lowercase words or capitalize a set of letters with simple keystroke sequences. To do so, use the following commands:

Keystrokes	Purpose
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.

Designating a Keystroke as a Command Entry

Sometimes you might want to use a particular keystroke as an executable command, perhaps as a shortcut. Use the following keystroke to insert a system code for this purpose:

Keystrokes
Press **Ctrl-V** or **Esc Q**.

Disabling and Reenabling Enhanced Editing Features

The above editing features were introduced in Cisco IOS Release 9.21, and are automatically enabled on your system. However, there may be some unique situations which could warrant disabling these enhanced editing features. For example, you may have prebuilt scripts that conflict with enhanced editing functionality. To globally disable enhanced editing mode and revert to the editing mode of software releases before Cisco IOS Release 9.21, use the following command in line configuration mode:

Command	Purpose
`no editing`	Disable the enhanced editing features for a particular line.

To disable enhanced editing mode for the current terminal session, use the following command in EXEC mode:

Command	Purpose
`terminal no editing`	Disable the enhanced editing features for the local line.

You can reenable enhanced editing mode with the **editing** command or **terminal editing** command.

To reenable the enhanced editing mode for the current terminal session, use the following command in EXEC mode:

Command	Purpose
`terminal editing`	Enable the enhanced editing features for the current terminal session.

To reconfigure a specific line to have enhanced editing mode, use the following command in line configuration mode:

Command	Purpose
`editing`	Enable the enhanced editing features.

Searching and Filtering CLI Output

The Cisco IOS CLI provides ways of searching through large amounts of command output, and filtering output to exclude information you do not need. These features are enabled for **show** and **more** commands, which generally display large amounts of data.

When output continues beyond what is displayed on your screen, the Cisco IOS CLI displays a **--More--** prompt. Pressing Return displays the next line; pressing the Spacebar displays the next screen of output. The Cisco IOS CLI String Search feature allows you to search or filter output from **--More--** prompts.

In addition to making information more manageable, using these features also provides the benefit of reducing router CPU usage by removing output prior to incurring the transmission costs.

The following sections explain and provide examples of using the CLI String Search feature:

- Understanding Regular Expressions
- Searching and Filtering show Commands
- Searching and Filtering more Commands
- Searching and Filtering from the --More-- Prompt
- Examples of Searching and Filtering

Understanding Regular Expressions

A regular expression is a pattern (a phrase, number, or more complex pattern) the CLI String Search feature matches against **show** or **more** command output. Regular expressions are case sensitive and allow for complex matching requirements. Simple regular expressions include entries like `Serial`, `misses`, or `138`. Complex regular expressions include entries like `00210...`, `( is )`, or `[Oo]utput`.

A regular expression can be a single-character pattern or a multiple-character pattern. That is, a regular expression can be a single character that matches the same single character in the command output or multiple characters that match the same multiple characters in the command output. The pattern in the command output is referred to as a string. This section describes creating both single-character patterns and multiple-character patterns. It also discusses creating more complex regular expressions using multipliers, alternation, anchoring, and parentheses.

Single-Character Patterns

The simplest regular expression is a single character that matches the same single character in the command output. You can use any letter (A-Z, a-z) or digit (0-9) as a single-character pattern. You can also use other keyboard characters (such as ! or ~) as single-character patterns, but certain keyboard characters have special meaning when used in regular expressions. Table 2 lists the keyboard characters with special meaning.

Table 2   Characters with Special Meaning

Character	Special Meaning
.	Matches any single character, including white space
*	Matchers 0 or more sequences of the pattern.
+	Matches 1 or more sequences of the pattern.
?	Matches 0 or 1 occurrences of the pattern.
^	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.

To use these special characters as single-character patterns, remove the special meaning by preceding each character with a backslash (\). The following examples are single-character patterns matching a dollar sign, an underscore, and a plus sign, respectively.

\$ _ \+

You can specify a range of single-character patterns to match against command output. For example, you can create a regular expression that matches a string containing one of the following letters: a, e, i, o, and u. One and only one of these characters must exist in the string for pattern matching to succeed. To specify a range of single-character patterns, enclose the single-character patterns in square brackets ([ ]). For example,

[aeiou]

matches any one of the five vowels of the lowercase alphabet, while

[abcdABCD]

matches any one of the first four letters of the lower- or uppercase alphabet.

You can simplify ranges by entering only the end points of the range separated by a dash (-). Simplify the previous range as follows:

[a-dA-D]

To add a dash as a single-character pattern in your range, include another dash and precede it with a backslash:

[a-dA-D\-]

You can also include a right square bracket (]) as a single-character pattern in your range. To do so, enter the following:

[a-dA-D\-\]]

The previous example matches any one of the first four letters of the lower- or uppercase alphabet, a dash, or a right square bracket.

You can reverse the matching of the range by including a caret (^) at the start of the range. The following example matches any letter except the ones listed.

[^a-dqsv]

The following example matches anything except a right square bracket (]) or the letter d:

[^\]d]

Multiple-Character Patterns

When creating regular expressions, you can also specify a pattern containing multiple characters. You create multiple-character regular expressions by joining letters, digits, or keyboard characters that do not have special meaning. For example, a4% is a multiple-character regular expression. Put a backslash in front of the keyboard characters that have special meaning when you want to remove their special meaning.

With multiple-character patterns, order is important. The regular expression a4% matches the character a followed by a 4 followed by a % sign. If the string does not have a4%, in that order, pattern matching fails. The following multiple-character regular expression

a.

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 the period character by putting a backslash in front of it. In the following expression

a\.

only the string a. matches this regular expression.

You can create a multiple-character regular expression containing all letters, all digits, all keyboard characters, or a combination of letters, digits, and other keyboard characters. The following examples are all valid regular expressions:

telebit 3107 v32bis

Multipliers

You can create more complex regular expressions that instruct Cisco IOS software to match multiple occurrences of a specified regular expression. To do so, you use some special characters with your single- and multiple-character patterns. Table 3 lists the special characters that specify "multiples" of a regular expression.

Table 3   Special Characters Used as Multipliers

Character	Description
*	Matches 0 or more single- or multiple-character patterns.
+	Matches 1 or more single- or multiple-character patterns.
?	Matches 0 or 1 occurrences of the single- or multiple-character pattern.

The following example matches any number of occurrences of the letter a, including none:

a*

The following pattern requires there to be at least one letter a in the string to be matched:

a+

The following pattern matches the string bb or bab:

ba?b

The following string matches any number of asterisks (*):

*

To use multipliers with multiple-character patterns, you enclose the pattern in parentheses. In the following example, the pattern matches any number of the multiple-character string ab:

(ab)*

As a more complex example, the following pattern matches one or more instances of alphanumeric pairs (but not none; that is, an empty string is not a match):

([A-Za-z][0-9])+

The order for matches using multipliers (*, +, or ?) is to put the longest construct first. Nested constructs are matched from outside to inside. Concatenated constructs are matched beginning at the left side of the construct. Thus, the regular expression matches A9b3, but not 9Ab3 because the letters are specified before the numbers.

Alternation

Alternation allows you to specify alternative patterns to match against a string. You separate the alternative patterns with a vertical bar (|). Exactly one of the alternatives can match the string. For example, the regular expression

codex|telebit

matches the string codex or the string telebit, but not both codex and telebit.

Anchoring

You can instruct Cisco IOS software to match a regular expression pattern against the beginning or the end of the string. That is, you can specify that the beginning or end of a string contain a specific pattern. You "anchor" these regular expressions to a portion of the string using the special characters shown in Table 4.

Table 4   Special Characters Used for Anchoring

Character	Description
^	Matches the beginning of the string.
$	Matches the end of the string.

There is another use for the ^ symbol. For example, the following regular expression matches an string only if the string starts with abcd:

^abcd

Whereas the following expression is in a range that matches any single letter, as long as it is not the letters a, b, c, or d:

[^abcd]

With the following example, the regular expression matches a string that ends with .12:

$\.12

Contrast these anchoring characters with the special character underscore (_). Underscore matches the beginning of a string (^), the end of a string ($), parentheses ( ) , space ( ), braces { }, comma (,), or underscore (_). With the underscore character, you can specify that a pattern exist anywhere in the string. For example,

_1300_

matches any string that has 1300 somewhere in the string. The string's 1300 can be preceded by or end with a space, brace, comma, or underscore. So, while

{1300_

matches the regular expression, 21300 and 13000 do not.

Using the underscore character, you can replace long regular expression lists, such as the following:

^1300$ ^1300(space) (space)1300 {1300, ,1300, {1300} ,1300, (1300

with simply _1300_.

Parentheses for Recall

As shown in the "Multipliers" section, you use parentheses with multiple-character regular expressions to multiply the occurrence of a pattern. You can also use parentheses around a single- or multiple-character pattern to instruct the Cisco IOS software to remember a pattern for use elsewhere in the regular expression.

To create a regular expression that recalls a previous pattern, you use parentheses to indicate memory of a specific pattern and a backslash (\) followed by an integer to re-use the remembered pattern. The integer specifies the occurrence of a parentheses in the regular expression pattern. If you have more than one remembered pattern in your regular expression, then \1 indicates the first remembered pattern, and \2 indicates the second remembered pattern, and so on.

The following regular expression uses parentheses for recall:

a(.)bc(.)\1\2

This regular expression matches an a followed by any character (call it character #1), followed by bc followed by any character (character #2), followed by character #1 again, followed by character #2 again. So, the regular expression can match aZbcTZT. The software remembers that character #1 is Z and character #2 is T and then uses Z and T again later in the regular expression.

Searching and Filtering show Commands

To search **show** command output, use the following command in EXEC mode:

Command	Purpose
show *any-command* | **begin** *regular-expression*	Begin unfiltered output of the **show** *any-command* with the first line that contains the regular expression.
Ctrl-^	Interrupt output.

To filter **show** command output, use one of the following commands in EXEC mode:

Command	Purpose
show *any-command* | **exclude** *regular-expression*	Display output lines that do not contain the regular expression.
show *any-command* | **include** *regular-expression*	Display output lines that contain the regular expression.
Ctrl-^	Interrupt output.

Note   A few **show** commands that have long output requirements use no responses at the --More-- prompt to jump to the next table of output; these outputs require you to enter the same number of Ctrl-^s as you would no responses to completely abort output.

Searching and Filtering more Commands

You can search **more** commands the same way you search **show** commands. To search **more** command output, use the following command in EXEC mode:

Command	Purpose
more *any-command* | **begin** *regular-expression*	Begin unfiltered output of the **more** *any-command* with the first line that contains the regular expression.
Ctrl-^	Interrupt output.

You can filter **more** commands the same way you filter **show** commands. To filter **more** command output, use one of the following commands in EXEC mode:

Command	Purpose
more *any-command* | **exclude** *regular-expression*	Display output lines that do not contain the regular expression.
more *any-command* | **include** *regular-expression*	Display output lines that contain the regular expression.
Ctrl-^	Interrupt output.

Searching and Filtering from the --More-- Prompt

You can search output from --More-- prompts. To search **show** or **more** command output from a --More-- prompt, use the following command in EXEC mode:

Command	Purpose
/regular-expression	Begin unfiltered output with the first line that contains the regular expression.
Ctrl-^	Interrupt output.

You can filter output from --More-- prompts. However, you can only specify one filter for each command's output. The filter remains until the **show** or **more** command output finishes or until you interrupt the output (using **Ctrl-^**).Therefore, you cannot add a second filter at a --More-- prompt if you already specified a filter at the original command or at a previous --More--prompt.

To filter **show** or **more** command output at a --More-- prompt, use one of the following commands in EXEC mode:

Command	Purpose
-regular-expression	Display output lines that do not contain the regular expression.
+regular-expression	Display output lines that contain the regular expression.
Ctrl-^	Interrupt output.

Examples of Searching and Filtering

The following is partial sample output of the **more nvram:startup-config | begin** command that begins unfiltered output with the first line that contain the regular expression "ip." At the --More-- prompt, the user specifies a filter to exclude output lines that contain the regular expression "ip."

```
router# more nvram:startup-config | begin ip
ip subnet-zero
ip domain-name cisco.com
ip name-server 198.92.30.32
ip name-server 171.69.2.132
!
isdn switch-type primary-5ess
.
.
.
interface Ethernet1
 ip address 5.5.5.99 255.255.255.0
 --More--
-ip
filtering...
 media-type 10BaseT
!
interface Serial0:23
 encapsulation frame-relay
 no keepalive
 dialer string 4001
 dialer-group 1
 isdn switch-type primary-5ess
 no fair-queue
```

The following is partial sample output of the **more nvram:startup-config | include** command. It only displays lines that contain the regular expression "ip."

```
router# more nvram:startup-config | include ip
ip subnet-zero
ip domain-name cisco.com
ip name-server 198.92.30.32
ip name-server 171.69.2.132
```

The following is partial sample output of the **more nvram:startup-config | exclude** command. It excludes lines that contain the regular expression "service." At the --More-- prompt, the user searches for the regular expression "Dialer1." This continues filtered output with the first line that contains "Dialer1."

```
router# more nvram:startup-config | exclude service
!
version 12.0
!
hostname router
!
boot system flash
no logging buffered
!
ip subnet-zero
ip domain-name cisco.com
.
.
.
```

```
--More--
/Dialer1
filtering...
interface Dialer1
 no ip address
 no ip directed-broadcast
 dialer in-band
 no cdp enable
```

The following is partial sample output of the **show interface | begin** command that begins unfiltered output with the first line that contains the regular expression "Ethernet." At the --More-- prompt, the user specifies a filter to include only the lines that contain the regular expression "Serial."

```
router# show interface | begin Ethernet
Ethernet0 is up, line protocol is up
Hardware is Lance, address is 0060.837c.6399 (bia 0060.837c.6399)
  Description: ip address is 172.1.2.14 255.255.255.0
  Internet address is 172.1.2.14/24
.
.
.
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
--More--
+Serial
filtering...
Serial1 is up, line protocol is up
Serial2 is up, line protocol is up
Serial3 is up, line protocol is down
Serial4 is down, line protocol is down
Serial5 is up, line protocol is up
Serial6 is up, line protocol is up
Serial7 is up, line protocol is up
```

The following is partial sample output of the **show buffers | exclude** command. It excludes lines that contain the regular expression "0 misses." At the --More-- prompt, the user searches for the regular expression "Serial0." This continues the filtered output with the first line that contains "Serial0."

```
router# show buffers | exclude 0 misses

Buffer elements:
     398 in free list (500 max allowed)
Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50):
     50 in free list (20 min, 150 max allowed)
     551 hits, 3 misses, 0 trims, 0 created
Big buffers, 1524 bytes (total 50, permanent 50):
     49 in free list (5 min, 150 max allowed)
Very Big buffers, 4520 bytes (total 10, permanent 10):
.
.
.
Huge buffers, 18024 bytes (total 0 permanent 0):
     0 in free list (0 min, 4 max allowed)
--More--
/Serial0
filtering...
Serial0 buffers, 1543 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
```

The following is partial sample output of the **show interface | include** command. It only displays lines that contain the regular expression "(is)." The parenthesis force the inclusion of the spaces before and after "is." This ensures that only lines containing "is" with a space both before and after it will be included in the output. This excludes lines with words like "disconnect."

```
router# show interface | include ( is )
ATM0 is administratively down, line protocol is down
  Hardware is ATMizer BX-50
Dialer1 is up (spoofing), line protocol is up (spoofing)
  Hardware is Unknown
  DTR is pulsed for 1 seconds on reset
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 0060.837c.6399 (bia 0060.837c.6399)
  Internet address is 172.21.53.199/24
Ethernet1 is up, line protocol is up
  Hardware is Lance, address is 0060.837c.639c (bia 0060.837c.639c)
  Internet address is 5.5.5.99/24
Serial0:0 is down, line protocol is down
  Hardware is DSX1
.
.
.
 --More--
```

At the --More-- prompt, the user searches for the regular expression "Serial0:13." This continues filtered output with the first line that contains "Serial0:13."

```
/Serial0:13
filtering...
Serial0:13 is down, line protocol is down
  Hardware is DSX1
  Internet address is 11.0.0.2/8
     0 output errors, 0 collisions, 2 interface resets
  Timeslot(s) Used:14, Transmitter delay is 0 flag
```