Basic Command-Line Interface Commands

This chapter describes the commands used to enter and exit the various Cisco IOS configuration command modes. It provides a description of help features, command-line interface (CLI) navigation commands, and the command history feature.

The CLI allows you to enter partial Cisco IOS configuration commands. The software recognizes a command when you enter enough characters of the command to uniquely identify it.

For user interface task information and examples, see the “Using the Command-Line Interface” chapter of the Release 12.2 Cisco IOS Configuration Fundamentals Configuration Guide.
disable

To exit privileged EXEC mode and return to user EXEC mode, or to exit to a lower privilege level, enter the `disable` EXEC command.

```
disable [privilege-level]
```

**Syntax Description**

| `privilege-level` | (Optional) Specific privilege level (other than user EXEC mode). |

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Up to 16 security levels can be configured using Cisco IOS software. If such levels are configured on a system, using this command with the `privilege-level` option allows you to exit to a lower security level. If a level is not specified, the user will exit to the user EXEC mode, which is the default.

***Note***

Five EXEC commands are associated with privilege level 0: `disable`, `enable`, `exit`, `help`, and `logout`. If you configure a privilege level greater than 0, these five commands will not be included in the command set for that privilege level.

**Examples**

In the following example, the user enters privileged EXEC mode using the `enable` command, then exits back to user EXEC mode using the `disable` command. Note that the prompt for user EXEC mode is `>`, and the prompt for privileged EXEC mode is `#`.

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

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>enable</code></td>
<td>Enables higher privilege level access, such as privileged EXEC mode.</td>
</tr>
</tbody>
</table>
To reenable Cisco IOS enhanced editing features for a particular line after they have been disabled, use the `editing` line configuration command. To disable these features, use the `no` form of this command.

```
editing
no editing
```

**Syntax Description**
This command has no arguments or keywords.

**Defaults**
Enabled

**Command Modes**
Line configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
Enhanced editing features are enabled by default. However, there may be situations in which you need to disable these features. The `no` form of this command disables these enhanced editing features, and the plain form of the command can be used to reenable these features.

Table 3 provides a description of the keys used to enter and edit commands when the editing features are enabled. Ctrl indicates the Control key, which must be pressed simultaneously with its associated letter key. Esc indicates the Escape key, which must be pressed first, followed by its associated letter key. A comma is used in the following table to indicate a key sequence (the comma key should not be pressed). Keys are not case sensitive. Many letters used for CLI navigation and editing were chosen to provide an easy way of remembering their functions. In the following table (Table 3), characters are bolded in the “Function Summary” column to indicate the relation between the letter used and the function.
### Table 3  Command Editing Keys and Functions

<table>
<thead>
<tr>
<th>Keys</th>
<th>Function Summary</th>
<th>Function Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tab</td>
<td>Complete command</td>
<td>Completes a partial command name entry. When you enter a unique set of characters and press the Tab key, the system completes the command name. If you enter a set of characters that could indicate more than one command, the system beeps to indicate an error. To view the commands which match the set of characters you have entered, enter a question mark (?) immediately following the partial command (no space). The CLI will then list the commands that begin with that string.</td>
</tr>
<tr>
<td>Return (at the command line)</td>
<td>Execute</td>
<td>Executes the command.</td>
</tr>
<tr>
<td>Return (at the --More--prompt)</td>
<td>Continue</td>
<td>Displays the next line of output.</td>
</tr>
<tr>
<td>Space Bar (at the --More--prompt)</td>
<td>Continue</td>
<td>Displays the next screen of output. The amount of output you see will depend on the screen depth setting of your terminal.</td>
</tr>
<tr>
<td>Delete or Backspace</td>
<td>Backspace</td>
<td>Erases the character to the left of the cursor.</td>
</tr>
<tr>
<td>Left Arrow(^1) or Ctrl-B</td>
<td>Back character</td>
<td>Moves the cursor one character to the left. When you enter a command that extends beyond a single line, you can press the Left Arrow or Ctrl-B keys repeatedly to scroll back toward the system prompt and verify the beginning of the command entry.</td>
</tr>
<tr>
<td>Right Arrow(^1) or Ctrl-F</td>
<td>Forward character</td>
<td>Moves the cursor one character to the right.</td>
</tr>
<tr>
<td>Esc, B</td>
<td>Back word</td>
<td>Moves the cursor back one word.</td>
</tr>
<tr>
<td>Esc, F</td>
<td>Forward word</td>
<td>Moves the cursor forward one word.</td>
</tr>
<tr>
<td>Ctrl-A</td>
<td>Beginning of line</td>
<td>Moves the cursor to the beginning of the line.</td>
</tr>
<tr>
<td>Ctrl-E</td>
<td>End of line</td>
<td>Moves the cursor to the end of the command line.</td>
</tr>
<tr>
<td>Ctrl-D</td>
<td>Delete character</td>
<td>Deletes the character at the cursor.</td>
</tr>
<tr>
<td>Esc, D</td>
<td>Delete next word</td>
<td>Deletes from the cursor to the end of the word.</td>
</tr>
<tr>
<td>Ctrl-W</td>
<td>Delete previous word</td>
<td>Deletes the word to the left of the cursor.</td>
</tr>
<tr>
<td>Ctrl-K</td>
<td>Delete line forward</td>
<td>Deletes all characters from the cursor to the end of the command line.</td>
</tr>
<tr>
<td>Ctrl-U or Ctrl-X</td>
<td>Delete line backward</td>
<td>Deletes all characters from the cursor back to the beginning of the command line.</td>
</tr>
<tr>
<td>Ctrl-T</td>
<td>Transpose characters</td>
<td>Transposes the character to the left of the cursor with the character located at the cursor.</td>
</tr>
</tbody>
</table>
Table 3  Command Editing Keys and Functions (continued)

<table>
<thead>
<tr>
<th>Keys</th>
<th>Function Summary</th>
<th>Function Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl-R or Ctrl-L</td>
<td>Redisplay line</td>
<td>Redisplays the system prompt and command line.</td>
</tr>
<tr>
<td>Ctrl-V or Esc, Q</td>
<td>Ignore editing</td>
<td>Inserts a code to indicate to the system that the keystroke immediately following should be treated as a command entry, not as an editing key.</td>
</tr>
<tr>
<td>Up Arrow(^1) or Ctrl-P</td>
<td>Previous command</td>
<td>Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.</td>
</tr>
<tr>
<td>Down Arrow(^1) or Ctrl-N (next)</td>
<td>Next command</td>
<td>Returns to more recent commands in the history buffer (after recalling commands with the Up Arrow or Ctrl-P). Repeat the key sequence to recall successively more recent commands.</td>
</tr>
<tr>
<td>Ctrl-Y</td>
<td>Recall last deleted command</td>
<td>Recalls the most recent entry in the delete buffer. The delete buffer contains the last ten items you have deleted or cut. Ctrl-Y can be used in conjunction with Esc Y.</td>
</tr>
<tr>
<td>Esc, Y</td>
<td>Recall next deleted command</td>
<td>Recalls the next entry in the delete buffer. The delete buffer contains the last ten items you have deleted. Press Ctrl-Y first to recall the most recent entry. Then press Esc Y up to nine times to recall the remaining entries in the buffer. If you bypass an entry, continue to press Esc Y to cycle back to it.</td>
</tr>
<tr>
<td>Esc, C</td>
<td>Capitalize word</td>
<td>Capitalizes the word from the cursor to the end of the word.</td>
</tr>
<tr>
<td>Esc, U</td>
<td>Make word uppercase</td>
<td>Changes all letters from the cursor to the next space on the line appear in uppercase letters.</td>
</tr>
<tr>
<td>Esc, L</td>
<td>Make word lowercase</td>
<td>Changes the word to lowercase from the cursor to the end of the word.</td>
</tr>
</tbody>
</table>

1. The arrow keys function only with ANSI-compatible terminals.

Examples

In the following example, enhanced editing mode is disabled on line 3:

```
Router(config)# line 3
Router(config-line)# no editing
```

Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>terminal editing</td>
<td>Controls CLI enhanced editing feature for the current terminal session.</td>
</tr>
</tbody>
</table>
enable

To enter privileged EXEC mode, or any other security level set by a system administrator, use the enable EXEC command.

```plaintext
enable [privilege-level]
```

**Syntax Description**

`privilege-level` (Optional) Privilege level at which to log in.

**Defaults**

Privilege-level 15 (privileged EXEC)

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Entering privileged EXEC mode enables the use of privileged commands. Because many of the privileged commands set operating parameters, privileged access should be password-protected to prevent unauthorized use. If the system administrator has set a password with the `enable password` global configuration command, you are prompted to enter it before being allowed access to privileged EXEC mode. The password is case sensitive.

If an `enable` password has not been set, enable mode only can be accessed through the console connection.

Security levels can be set by an administrator using the `enable password` and `privilege level` commands. Up to 16 privilege levels can be specified, using the numbers 0 through 15. Using these privilege levels, the administrator can allow or deny access to specific commands. Privilege level 0 is associated with user EXEC mode, and privilege level 15 is associated with privileged EXEC mode.

For more information on defined privilege levels, see the “Passwords and Privileges” chapters of the *Cisco IOS Security Configuration Guide* and the *Cisco IOS Security Command Reference* publications.

If a level is not specified when entering the `enable` command, the user will enter the default mode of privileged EXEC (level 15).
Examples

In the following example, the user enters privileged EXEC mode using the `enable` command. The system prompts the user for a password before allowing access to the privileged EXEC mode. The password is not printed to the screen. The user then exits back to user EXEC mode using the `disable` command. Note that the prompt for user EXEC mode is `>`, and the prompt for privileged EXEC mode is `#`.

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

Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>disable</code></td>
<td>Exits from privileged EXEC mode to user EXEC mode, or, if privilege levels are set, to the specified privilege level.</td>
</tr>
<tr>
<td><code>enable password</code></td>
<td>Sets a local password to control access to various privilege levels.</td>
</tr>
<tr>
<td><code>privilege level (global)</code></td>
<td>Sets a privilege level for a command.</td>
</tr>
<tr>
<td><code>privilege level (line)</code></td>
<td>Sets a privilege level for a command for a specific line.</td>
</tr>
</tbody>
</table>
end

To end the current configuration session and return to privileged EXEC mode, use the `end` global configuration command.

```
end
```

### Syntax Description
This command has no arguments or keywords.

### Defaults
No default behavior or values

### Command Modes
Global configuration

### Command History
```
<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>
```

### Usage Guidelines
This command will bring you back to privileged EXEC mode regardless of what configuration mode or configuration submode you are in.

**Note**

This global configuration command can be used in any configuration mode.

Use this command when you are done configuring the system and you want to return to EXEC mode to perform verification steps.

### Examples
In the following example, the `end` command is used to exit from ALPS ASCU configuration mode and return to privileged EXEC mode. A `show` command is used in privileged EXEC mode to verify the configuration.

```
Router# configure terminal
Router(config)# interface serial 1:1
Router(config-if)# alps ascu 4B
Router(config-alps-ascu)# end
Router# show interface serial 1:1
```

### Related Commands
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>exit (global)</td>
<td>Exits from the current configuration mode.</td>
</tr>
</tbody>
</table>
exit (EXEC)

To close an active terminal session by logging off the router, use the `exit` command in EXEC mode.

```
exit
```

**Syntax Description**

This command has no arguments or keywords.

**Defaults**

No default behavior or values

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `exit` (EXEC) command in EXEC mode to exit the active session (log off the device).

**Examples**

In the following example, the `exit` (global) command is used to move from global configuration mode to privileged EXEC mode, the `disable` command is used to move from privileged EXEC mode to user EXEC mode, and the `exit` (EXEC) command is used to log off (exit the active session):

```
Router(config)# exit
Router# disable
Router> exit
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>disconnect</td>
<td>Disconnects a line.</td>
</tr>
<tr>
<td>end</td>
<td>Exits configuration mode, or any of the configuration submodes.</td>
</tr>
<tr>
<td>exit (global)</td>
<td>Exits from the current configuration mode to the next highest configuration mode.</td>
</tr>
</tbody>
</table>
exit (global)

To exit any configuration mode to the next highest mode in the CLI mode hierarchy, use the `exit` command in any configuration mode.

```
exit
```

**Syntax Description**

This command has no arguments or keywords.

**Defaults**

No default behavior or values

**Command Modes**

All configuration modes

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `exit` command is used in the Cisco IOS CLI to exit from the current command mode to the next highest command mode in the CLI mode hierarchy.

For example, use the `exit` command in global configuration mode to return to privileged EXEC mode. Use the `exit` command in interface, line, or router configuration mode to return to global configuration mode. Use the `exit` command in subinterface configuration mode to return to interface configuration mode. At the highest level, EXEC mode, the `exit` command will exit the EXEC mode and disconnect from the router interface (see the description of the `exit (EXEC)` command for details).

**Examples**

The following example displays an exit from the subinterface configuration mode to return to the interface configuration mode:

```
Router(config-subif)# exit
Router(config-if)#
```

The following example displays an exit from the interface configuration mode to return to the global configuration mode:

```
Router(config-if)# exit
Router(config)#
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>disconnect</code></td>
<td>Disconnects a line.</td>
</tr>
<tr>
<td><code>end</code></td>
<td>Exits from any configuration mode to privileged EXEC mode.</td>
</tr>
<tr>
<td><code>exit (EXEC)</code></td>
<td>Closes the active terminal session by logging off the router.</td>
</tr>
</tbody>
</table>
**full-help**

To get help for the full set of user-level commands, use the `full-help` line configuration command.

```
full-help
```

**Syntax Description**

This command has no arguments or keywords.

**Defaults**

Disabled

**Command Modes**

Line configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `full-help` command enables (or disables) an unprivileged user to see all of the help messages available. It is used with the `show ?` command.

**Examples**

In the following example, the `show ?` command is used first with full-help disabled. Then full-help is enabled for the line, and the `show ?` command is used again to demonstrate the additional help output that is displayed.

```
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
diagnosis  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> enable
Password:<letmein>

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```
Router(config)# **line** console 0
Router(config-line)# **full-help**
Router(config-line)# ^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router# disable
Router> show ?

<table>
<thead>
<tr>
<th>access-expression</th>
<th>List access expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>access-lists</td>
<td>List access lists</td>
</tr>
<tr>
<td>aliases</td>
<td>Display alias commands</td>
</tr>
<tr>
<td>apollo</td>
<td>Apollo network information</td>
</tr>
<tr>
<td>appltalk</td>
<td>AppleTalk information</td>
</tr>
<tr>
<td>arp</td>
<td>ARP table</td>
</tr>
<tr>
<td>async</td>
<td>Information on terminal lines used as router interfaces</td>
</tr>
<tr>
<td>bootflash</td>
<td>Boot Flash information</td>
</tr>
<tr>
<td>bridge</td>
<td>Bridge Forwarding/Filtering Database [verbose]</td>
</tr>
<tr>
<td>bsc</td>
<td>BSC interface information</td>
</tr>
<tr>
<td>bstun</td>
<td>BSTUN interface information</td>
</tr>
<tr>
<td>buffers</td>
<td>Buffer pool statistics</td>
</tr>
<tr>
<td>calendar</td>
<td>Display the hardware calendar</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>translate</td>
<td>Protocol translation information</td>
</tr>
<tr>
<td>ttycap</td>
<td>Terminal capability tables</td>
</tr>
<tr>
<td>users</td>
<td>Display information about terminal lines</td>
</tr>
<tr>
<td>version</td>
<td>System hardware and software status</td>
</tr>
<tr>
<td>vines</td>
<td>VINES information</td>
</tr>
<tr>
<td>vlans</td>
<td>Virtual LANs Information</td>
</tr>
<tr>
<td>whoami</td>
<td>Info on current tty line</td>
</tr>
<tr>
<td>x25</td>
<td>X.25 information</td>
</tr>
<tr>
<td>xns</td>
<td>XNS information</td>
</tr>
<tr>
<td>xremote</td>
<td>XRemote statistics</td>
</tr>
</tbody>
</table>

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>Displays a brief description of the help system.</td>
</tr>
</tbody>
</table>
help

To display a brief description of the help system, enter the `help` command.

```text
help
```

**Syntax Description**

This command has no arguments or keywords.

**Defaults**

No default behavior or values

**Command Modes**

All command modes

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `help` command provides a brief description of the context-sensitive help system, which functions as follows:

- To list all commands available for a particular command mode, enter a question mark (`?`) at the system prompt.
- To obtain a list of commands that begin with a particular character string, enter the abbreviated command entry immediately followed by a question mark (`?`). This form of help is called *word help*, because it lists only the keywords or arguments that begin with the abbreviation you entered.
- To list the keywords and arguments associated with a command, enter a question mark (`?`) in place of a keyword or argument on the command line. This form of help is called *command syntax help*, because it lists the keywords or arguments that apply based on the command, keywords, and arguments you have already entered.

**Examples**

In the following example, the `help` command is used to display a brief description of the help system:

```text
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 backup 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?'`).
The following example shows how to use word help to display all the privileged EXEC commands that begin with the letters “co”. The letters entered before the question mark are reprinted on the next command line to allow the user to continue entering the command.

Router# co?
configure connect copy
Router# co

The following example shows how to use command syntax help to display the next argument of a partially complete access-list command. One option is to add a wildcard mask. The <cr> symbol indicates that the other option is to press Return to execute the command without adding any more keywords or arguments. The characters entered before the question mark are reprinted on the next command line to allow the user to continue entering the command or to execute that command as it is.

Router(config)# access-list 99 deny 131.108.134.234 ?
A.B.C.D Mask of bits to ignore
<cr>
Router(config)# access-list 99 deny 131.108.134.234

<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>full-help</td>
<td></td>
<td>Gets help for the full set of user-level commands.</td>
</tr>
</tbody>
</table>
**history**

To enable the command history function, use the `history` line configuration command. To disable the command history feature, use the `no` form of this command.

```
history
no history
```

**Syntax Description**

This command has no arguments or keywords.

**Defaults**

Enabled, ten command lines in buffer

**Command Modes**

Line configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The command history feature provides a record of EXEC commands that you have entered. This feature is particularly useful for recalling long or complex commands or entries, including access lists.

To change the number of command lines that the system will record in its history buffer, use the `history size` line configuration command.

The `history` command enables the history function with the last buffer size specified or, if there was not a prior setting, with the default of ten lines. The `no history` command disables the history feature.

The `show history` EXEC command will list the commands you have entered, but you can also use your keyboard to display individual commands. Table 4 lists the keys you can use to recall commands from the command history buffer.

**Table 4 History Keys**

<table>
<thead>
<tr>
<th>Key(s)</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl-P or Up Arrow&lt;sup&gt;1&lt;/sup&gt;</td>
<td>Recalls commands in the history buffer in a backward sequence, beginning with the most recent command. Repeat the key sequence to recall successively older commands.</td>
</tr>
<tr>
<td>Ctrl-N or Down Arrow&lt;sup&gt;1&lt;/sup&gt;</td>
<td>Returns to more recent commands in the history buffer after recalling commands with Ctrl-P or the Up Arrow. Repeat the key sequence to recall successively more recent commands.</td>
</tr>
</tbody>
</table>

<sup>1</sup> The arrow keys function only with ANSI-compatible terminals.

**Examples**

In the following example, line 4 is configured with a history buffer size of 35 lines:

```
Router(config)# line 4
```
Router(config-line)# **history size 35**

<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>history size</strong></td>
<td>Sets the command history buffer size for a particular line.</td>
</tr>
<tr>
<td></td>
<td><strong>show history</strong></td>
<td>Lists the commands you have entered in the current EXEC session.</td>
</tr>
<tr>
<td></td>
<td><strong>terminal history</strong></td>
<td>Enables the command history feature for the current terminal session or changes the size of the command history buffer for the current terminal session.</td>
</tr>
</tbody>
</table>
history size

To change the command history buffer size for a particular line, use the `history size` line configuration command. To reset the command history buffer size to ten lines, use the `no` form of this command.

```
history size number-of-lines

no history size
```

**Syntax Description**

- `number-of-lines`: Specifies the number of command lines that the system will record in its history buffer. The range is from 0 to 256. The default is ten.

**Defaults**

Ten command lines

**Command Modes**

Line configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `history size` command should be used in conjunction with the `history` and `show history` commands. The `history` command enables or disables the command history function. The `show history` command lists the commands you have entered in the current EXEC session. The number of commands that the history buffer will show is set by the `history size` command.

**Note**

The `history size` command only sets the size of the buffer; it does not reenable the history feature. If the `no history` command is used, the `history` command must be used to reenable this feature.

**Examples**

The following example displays line 4 configured with a history buffer size of 35 lines:

```
Router(config)# line 4
Router(config-line)# history size 35
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>history</td>
<td>Enables or disables the command history function.</td>
</tr>
<tr>
<td>show history</td>
<td>Lists the commands you have entered in the current EXEC session.</td>
</tr>
<tr>
<td>terminal history size</td>
<td>Enables the command history function for the current terminal session or changes the size of the command history buffer for the current terminal session.</td>
</tr>
</tbody>
</table>
logout

To close an active terminal session by logging off the router, use the `logout` command in user EXEC mode.

```
logout
```

**Syntax Description**

This command has no arguments or keywords.

**Defaults**

No default behavior or values.

**Command Modes**

User EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Examples**

In the following example, the `exit` (global) command is used to move from global configuration mode to privileged EXEC mode, the `disable` command is used to move from privileged EXEC mode to user EXEC mode, and the `logout` command is used to log off (exit from the active session):

```
Router(config)# exit
Router# disable
Router> logout
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>exit</code> (global)</td>
<td>Exits any configuration mode to the next highest mode in the CLI mode hierarchy.</td>
</tr>
</tbody>
</table>
menu (EXEC)

To display a preconfigured user menu, use the `menu` command in user or privileged EXEC mode.

```
menu menu-name
```

**Syntax Description**
- `menu-name` The name of the menu.

**Command Modes**
- EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

A user menu is a type of user interface where text descriptions of actions to be performed are displayed to the user. The user can use the menu to select services and functions without having to know the details of command-line interface (CLI) commands.

Menus can be created for users in global configuration mode, using the commands listed in the “Related Commands” section. The description of these commands can be found in the “Connection, Menu, and System Banner Commands” chapter of this document.

A menu can be invoked at either the user or privileged EXEC level, but if an item in the menu contains a privileged EXEC command, the user must be logged in at the privileged level for the command to succeed.

**Examples**

The following example invokes a menu named OnRamp:

```
Router> menu OnRamp
Welcome to OnRamp Internet Services
Type a number to select an option;
    Type 9 to exit the menu.
1     Read email
2     UNIX Internet access
3     Resume UNIX connection
6     Resume next connection
9     Exit menu system
```
## Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>menu clear-screen</td>
<td>Clears the terminal screen before displaying a menu.</td>
</tr>
<tr>
<td>menu command</td>
<td>Specifies underlying commands for user interface menus.</td>
</tr>
<tr>
<td>menu default</td>
<td>Specifies the menu item to use as the default.</td>
</tr>
<tr>
<td>menu line-mode</td>
<td>Requires the user to press Enter after specifying an option number.</td>
</tr>
<tr>
<td>menu options</td>
<td>Sets options for items in user interface menus.</td>
</tr>
<tr>
<td>menu prompt</td>
<td>Specifies the prompt for a user interface menu.</td>
</tr>
<tr>
<td>menu single-space</td>
<td>Displays menu items single-spaced rather than double-spaced.</td>
</tr>
<tr>
<td>menu status-line</td>
<td>Displays a line of status information about the current user at the top of a menu</td>
</tr>
<tr>
<td>menu text</td>
<td>Specifies the text of a menu item in a user interface menu.</td>
</tr>
<tr>
<td>menu title</td>
<td>Creates a title, or banner, for a user menu.</td>
</tr>
<tr>
<td>no menu</td>
<td>Deletes a specified menu from a menu configuration.</td>
</tr>
</tbody>
</table>
more begin

To search the output of any `more` command, use the `more begin` command in EXEC mode. This command begins unfiltered output of the `more` command with the first line that contains the regular expression you specify.

```
more file-url | begin regular-expression
```

**Syntax Description**

| File-url | The Universal Resource Locator (url) of the file to display. More commands are advanced show commands; for details, see the command reference page in this book for the `more` command.
|---|---
| | A vertical bar (the “pipe” symbol) indicates that an output processing specification follows.
| Regular-expression | Any regular expression found in `more` command output.
| / | Specifies a search at a --More-- prompt that begins unfiltered output with the first line that contains the regular expression.
| - | Specifies a filter at a --More-- prompt that only displays output lines that do not contain the regular expression.
| + | Specifies a filter at a --More-- prompt that only displays output lines that contain the regular expression.

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.3 AA</td>
<td>The <code>more</code> command was introduced.</td>
</tr>
<tr>
<td>12.0(1)T</td>
<td>This extension of the <code>more</code> command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `regular-expression` argument is case sensitive and allows for complex matching requirements. You can specify a new search at every --More-- prompt.

To search the remaining output of the `more` command, use the following command at the --More-- prompt:

```
/regular-expression
```

To filter the remaining output of the `more` command, use one of the following commands at the --More-- prompt:

```
-regular-expression
+regular-expression
```

When output volume is large, the search can produce long lists of output. To interrupt the output, press `Ctrl-^` (Ctrl-Shift-6) or `Ctrl-Z`. 

Once you specify a filter for a `more` command, you cannot specify another filter at a `--More--` prompt. The first specified filter remains until the `more` command output finishes or until you interrupt the output. The use of the keyword `begin` does not constitute a filter.

Because prior output is not saved, you cannot search or filter backward through prior output.

### Examples

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
```

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>more exclude</code></td>
<td>Filters <code>more</code> command output so that it excludes lines that contain a particular regular expression.</td>
</tr>
<tr>
<td><code>more include</code></td>
<td>Filters <code>more</code> command output so that it displays only lines that contain a particular regular expression.</td>
</tr>
<tr>
<td><code>show begin</code></td>
<td>Searches the output of any <code>show</code> command and displays the output from the first instance of a specified string.</td>
</tr>
<tr>
<td><code>show exclude</code></td>
<td>Filters <code>show</code> command output so that it excludes lines that contain a particular regular expression.</td>
</tr>
<tr>
<td><code>show include</code></td>
<td>Filters <code>show</code> command output so that it displays only lines that contain a particular regular expression.</td>
</tr>
</tbody>
</table>
more exclude

To filter more command output so that it excludes lines that contain a particular regular expression, use the more exclude command in EXEC mode.

```
more file-url | exclude regular-expression
```

**Syntax Description**

- `file-url` The Universal Resource Locator (url) of the file to display. More commands are advanced show commands; for details, see the command reference page in this book for the more command.
- `|` A vertical bar (the “pipe” symbol) indicates that an output processing specification follows.
- `regular-expression` Any regular expression found in more command output.
- `/` Specifies a search at a --More-- prompt that begins unfiltered output with the first line that contains the regular expression.

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.3 AA</td>
<td>The more command was introduced.</td>
</tr>
<tr>
<td>12.0(1)T</td>
<td>This extension of the more command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The regular-expression argument is case sensitive and allows for complex matching requirements.

You can specify a new search at any --More-- prompt. To search the remaining output of the more command, use the following command at the --More-- prompt:

```
/regular-expression
```

When output volume is large, the search can produce long lists of output. To interrupt the output, press Ctrl-^ (Ctrl-Shift-6) or Ctrl-Z.

Because prior output is not saved, you cannot search or filter backward through prior output.

**Examples**

The following is partial sample output of the more nvram:startup-config | exclude command. The use of | exclude service in the command specifies a filter that excludes lines that contain the regular expression “service.” At the --More-- prompt, the user searches for the regular expression “Dialer1,” which 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>more begin</td>
<td>Begins unfiltered output of the more command with the first line that</td>
</tr>
<tr>
<td></td>
<td>contains the regular expression you specify.</td>
</tr>
<tr>
<td>more include</td>
<td>Filters more command output so that it displays only lines that contain a</td>
</tr>
<tr>
<td></td>
<td>particular regular expression.</td>
</tr>
<tr>
<td>show begin</td>
<td>Searches the output of any show command and displays the output from the</td>
</tr>
<tr>
<td></td>
<td>first instance of a specified string.</td>
</tr>
<tr>
<td>show exclude</td>
<td>Filters show command output so that it excludes lines that contain a</td>
</tr>
<tr>
<td></td>
<td>particular regular expression.</td>
</tr>
<tr>
<td>show include</td>
<td>Filters show command output so that it displays only lines that contain a</td>
</tr>
<tr>
<td></td>
<td>particular regular expression.</td>
</tr>
</tbody>
</table>
**more include**

To filter `more` command output so that it displays only lines that contain a particular regular expression, use the `more include` command in EXEC mode.

```
more file-url | include regular-expression
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>file-url</code></td>
<td>The Universal Resource Locator (url) of the file to display. More commands are advanced show commands; for details, see the command reference page in this book for the <code>more</code> command.</td>
</tr>
<tr>
<td>`</td>
<td>`</td>
</tr>
<tr>
<td><code>regular-expression</code></td>
<td>Any regular expression found in <code>more</code> command output.</td>
</tr>
<tr>
<td><code>/</code></td>
<td>Specifies a search at a --More-- prompt that begins unfiltered output with the first line that contains the regular expression.</td>
</tr>
</tbody>
</table>

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.3 AA</td>
<td>The <code>more</code> command was introduced.</td>
</tr>
<tr>
<td>12.0(1)T</td>
<td>This extension of the <code>more</code> command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `regular-expression` argument is case sensitive and allows for complex matching requirements.

You can specify a new search at any --More-- prompt. To search the remaining output of the `more` command, use the following syntax at the --More-- prompt:

```
/regular-expression
```

When output volume is large, the search can produce long lists of output. To interrupt the output, press `Ctrl^-` (Ctrl-Shift-6) or `Ctrl-Z`.

Because prior output is not saved, you cannot search or filter backward through prior output.

**Examples**

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
description ip address 172.21.53.199 255.255.255.0
ip address 172.21.53.199 255.255.255.0
```
### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>more begin</td>
<td>Begins unfiltered output of the more command with the first line that contains the regular expression you specify.</td>
</tr>
<tr>
<td>more exclude</td>
<td>Filters more command output so that it excludes lines that contain a particular regular expression.</td>
</tr>
<tr>
<td>show begin</td>
<td>Searches the output of any show command and displays the output from the first instance of a specified string.</td>
</tr>
<tr>
<td>show exclude</td>
<td>Filters show command output so that it excludes lines that contain a particular regular expression.</td>
</tr>
<tr>
<td>show include</td>
<td>Filters show command output so that it displays only lines that contain a particular regular expression.</td>
</tr>
</tbody>
</table>
show begin

To begin the output of any show command from a specified string, use the show begin command in EXEC mode.

```
show any-command | begin regular-expression
```

**Syntax Description**

<table>
<thead>
<tr>
<th>any-command</th>
<th>Any supported show command.</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>A vertical bar (the “pipe” symbol) indicates that an output processing specification follows.</td>
</tr>
<tr>
<td>regular-expression</td>
<td>Any regular expression found in show command output. The show output will begin from the first instance of this string (output prior to this string will not be printed to the screen). The string is case-sensitive. Use parenthesis to indicate a literal use of spaces.</td>
</tr>
<tr>
<td>/</td>
<td>Specifies a search at a --More-- prompt that begins unfiltered output with the first line that contains the regular expression.</td>
</tr>
<tr>
<td>-</td>
<td>Specifies a filter at a --More-- prompt that only displays output lines that do not contain the regular expression.</td>
</tr>
<tr>
<td>+</td>
<td>Specifies a filter at a --More-- prompt that only displays output lines that contain the regular expression.</td>
</tr>
</tbody>
</table>

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.3</td>
<td>The show command was introduced.</td>
</tr>
<tr>
<td>12.0(1)T</td>
<td>This extension of the show command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The regular-expression argument is case sensitive and allows for complex matching requirements. Use parenthesis to indicate a literal use of spaces. For example, `I begin u` indicates that the show output should begin with any line that contains a u; `I begin ( u)` indicates that the show output should begin with any line that contains a space and a u together (line has a word that begins with a lowercase u).

To search the remaining output of the show command, use the following command at the --More-- prompt:

```
/regular-expression
```

You can specify a filtered search at any --More-- prompt. To filter the remaining output of the show command, use one of the following commands at the --More-- prompt:

```
-regular-expression
+regular-expression
```

When output volume is large, the search can produce long lists of output. To interrupt the output, press Ctrl-^ (Ctrl-Shift-6) or Ctrl-z.
Once you specify a filter for a `show` command, you cannot specify another filter at the next `--More--` prompt. The first specified filter remains until the `more` command output finishes or until you interrupt the output. The use of the keyword `begin` does not constitute a filter.

Because prior output is not saved, you cannot search or filter backward through prior output.

A few `show` commands that have long output requirements do not require user input at the `--More--` prompt to jump to the next table of output; these types of output require you to enter the same number of Ctrl-^ or Ctrl-Z combinations as there are `--More--` prompts to completely abort output.

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 show only the lines in the remaining output 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
```

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>more begin</code></td>
<td>Begins unfiltered output of the <code>more</code> command with the first line that contains the regular expression you specify.</td>
</tr>
<tr>
<td><code>more exclude</code></td>
<td>Filters <code>more</code> command output so that it excludes lines that contain a particular regular expression.</td>
</tr>
<tr>
<td><code>more include</code></td>
<td>Filters <code>more</code> command output so that it displays only lines that contain a particular regular expression.</td>
</tr>
<tr>
<td><code>show exclude</code></td>
<td>Filters <code>show</code> command output so that it excludes lines that contain a particular regular expression.</td>
</tr>
<tr>
<td><code>show include</code></td>
<td>Filters <code>show</code> command output so that it displays only lines that contain a particular regular expression.</td>
</tr>
</tbody>
</table>
show exclude

To filter show command output so that it excludes lines that contain a particular regular expression, use the show exclude command in EXEC mode.

```
show any-command | exclude regular-expression
```

**Syntax Description**

<table>
<thead>
<tr>
<th>any-command</th>
<th>Any supported show command.</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>A vertical bar (the “pipe” symbol) indicates that an output processing specification follows.</td>
</tr>
<tr>
<td>regular-expression</td>
<td>Any regular expression found in show command output.</td>
</tr>
<tr>
<td>/</td>
<td>Specifies a search at a --More-- prompt that begins unfiltered output with the first line that contains the regular expression.</td>
</tr>
</tbody>
</table>

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.3</td>
<td>The show command was introduced.</td>
</tr>
<tr>
<td>T2.0(1)T</td>
<td>This extension of the show command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The regular-expression argument is case sensitive and allows for complex matching requirements. You can specify a new search at every --More-- prompt. To search the remaining output of the show command, use the following syntax at the --More-- prompt:

```
/regular-expression
```

When output volume is large, the search can produce long lists of output. To interrupt the output, press Ctrl-^ (Ctrl-Shift-6) or Ctrl-Z.

Because prior output is not saved, you cannot search or filter backward through prior output.

**Note**

A few show commands that have long output requirements do not require user input at the --More-- prompt to jump to the next table of output; these types of output require you to enter the same number of Ctrl-^ or Ctrl-Z combinations as there are --More-- prompts to completely abort output.

**Examples**

The following is partial sample output of the show | exclude command used with the show buffers command. It excludes lines that contain the regular expression “0 misses.” At the --More-- prompt, the user searches for the regular expression “Serial0,” which 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>more begin</td>
<td>Begins unfiltered output of the more command with the first line that</td>
</tr>
<tr>
<td></td>
<td>contains the regular expression you specify.</td>
</tr>
<tr>
<td>more exclude</td>
<td>Filters more command output so that it excludes lines that contain a</td>
</tr>
<tr>
<td></td>
<td>particular regular expression.</td>
</tr>
<tr>
<td>more include</td>
<td>Filters more command output so that it displays only lines that contain a</td>
</tr>
<tr>
<td></td>
<td>particular regular expression.</td>
</tr>
<tr>
<td>show begin</td>
<td>Searches the output of any show command and displays the output from the</td>
</tr>
<tr>
<td></td>
<td>first instance of a specified string.</td>
</tr>
<tr>
<td>show include</td>
<td>Filters show command output so that it displays only lines that contain a</td>
</tr>
<tr>
<td></td>
<td>particular regular expression.</td>
</tr>
</tbody>
</table>
show history

To list the commands you have entered in the current EXEC session, use the show history EXEC command.

    show history

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines

The command history feature provides a record of EXEC commands you have entered. The number of commands that the history buffer will record is determined by the history size line configuration command or the terminal history size EXEC command.

Table 5 lists the keys and functions you can use to recall commands from the command history buffer.

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl-P or Up Arrow¹</td>
<td>Recalls commands in the history buffer in a backward sequence, beginning with the most recent command. Repeat the key sequence to recall successively older commands.</td>
</tr>
<tr>
<td>Ctrl-N or Down Arrow¹</td>
<td>Returns to more recent commands in the history buffer after recalling commands with Ctrl-P or the Up Arrow. Repeat the key sequence to recall successively more recent commands.</td>
</tr>
</tbody>
</table>

1. The arrow keys function only with ANSI-compatible terminals.

Examples

The following is sample output from the show history command, which lists the commands the user has entered in EXEC mode for this session:

    Router# show history
    help
    where
    show hosts
    show history
    Router#
### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>history size</code></td>
<td>Enables the command history function, or changes the command history buffer size for a particular line.</td>
</tr>
<tr>
<td><code>terminal history size</code></td>
<td>Enables the command history feature for the current terminal session, or changes the size of the command history buffer for the current terminal session.</td>
</tr>
</tbody>
</table>
show include

To filter show command output so that it only displays lines that contain a particular regular expression, use the show include command in EXEC mode.

`show any-command | include regular-expression`

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>any-command</td>
<td>Any supported show command.</td>
</tr>
<tr>
<td>I</td>
<td>A vertical bar (the “pipe” symbol) indicates that an output processing</td>
</tr>
<tr>
<td></td>
<td>specification follows.</td>
</tr>
<tr>
<td>regular-expression</td>
<td>Any regular expression found in show command output. Use parenthesis</td>
</tr>
<tr>
<td></td>
<td>to include spaces in the expression.</td>
</tr>
<tr>
<td>/</td>
<td>Specifies a search at a --More-- prompt that begins unfiltered output with</td>
</tr>
<tr>
<td></td>
<td>the first line that contains the regular expression.</td>
</tr>
</tbody>
</table>

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.3</td>
<td>The show command was introduced.</td>
</tr>
<tr>
<td>12.0(1)T</td>
<td>This extension of the show command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `regular-expression` argument is case sensitive and allows for complex matching requirements.

You can specify a new search at every --More-- prompt. To search the remaining output of the show command, use the following syntax at the --More-- prompt:

`Irregular-expression`

When output volume is large, the search can produce long lists of output. To interrupt the output, press Ctrl-^ (Ctrl-Shift-6) or Ctrl-Z.

Because prior output is not saved, you cannot search or filter backward through prior output.

**Note**

A few show commands that have long output requirements do not require user input at the --More-- prompt to jump to the next table of output; these types of output require you to enter the same number of Ctrl-^ or Ctrl-Z combinations as there are --More-- prompts to completely abort output.

**Examples**

The following is partial sample output of the `show interface | include` command. It displays only lines that contain the regular expression“( is ).” The parentheses force the inclusion of the spaces before and after “is.” Use of the parenthesis ensures that only lines containing “is” with a space both before and after it will be included in the output. Lines with words like “disconnect” will be excluded because there are not spaces around the instance of the string “is”.

```
router# show interface | include ( is )
```
show include

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”, which 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 flags

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>more begin</td>
<td>Begins unfiltered output of the more command with the first line that contains the regular expression you specify.</td>
</tr>
<tr>
<td>more exclude</td>
<td>Filters more command output so that it excludes lines that contain a particular regular expression.</td>
</tr>
<tr>
<td>more include</td>
<td>Filters more command output so that it displays only lines that contain a particular regular expression.</td>
</tr>
<tr>
<td>show begin</td>
<td>Searches the output of any show command and displays the output from the first instance of a specified string.</td>
</tr>
<tr>
<td>show exclude</td>
<td>Filters show command output so that it excludes lines that contain a particular regular expression.</td>
</tr>
</tbody>
</table>
**terminal editing**

To reenable the enhanced editing mode for only the current terminal session, use the `terminal editing` EXEC command. To disable the enhanced editing mode on the current line, use the `no` form of this command.

```
  terminal editing

  terminal no editing
```

**Syntax Description**

This command has no arguments or keywords.

**Defaults**

Enabled

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

This command is identical to the `editing` EXEC mode command, except that it controls (enables or disables) enhanced editing for only the terminal session you are using. For a description of the available editing keys, see the description of the `editing` command in this chapter.

**Examples**

In the following example, enhanced editing mode is reenabled for only the current terminal session:

```
Router> terminal editing
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>editing</code></td>
<td>Controls CLI enhanced editing features for a particular line.</td>
</tr>
</tbody>
</table>
**terminal full-help**

To get help for the full set of user-level commands, use the `terminal full-help` EXEC mode command.

```
terminal full-help
```

**Syntax Description**
This command has no arguments or keywords.

**Defaults**
Disabled

**Command Modes**
EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
The `terminal full-help` command enables a user to see all of the help messages available from the terminal. It is used with the `show ?` command.

**Examples**

In the following example, the difference between the output of the `show ?` command before and after using the `terminal full-help` command is shown:

```
Router> show ?

access-expression  List access expression
access-lists       List access lists
aliases            Display alias commands
apollo             Apollo network information
bootflash          Boot Flash information
calendar           Display the hardware calendar
clock              Display the system clock
c 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
```
### Basic Command-Line Interface Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>full-help</td>
<td>Gets help for the full set of user-level commands.</td>
</tr>
<tr>
<td>help</td>
<td>Displays a brief description of the help system.</td>
</tr>
</tbody>
</table>

### Related Commands

- `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
- `compress`: Show compression statistics.
- `x25`: X.25 information
- `xns`: XNS information
- `xremote`: XRemote statistics
**terminal history**

To enable the command history feature for the current terminal session, use the `terminal history` command in user EXEC mode or privileged EXEC mode. To disable the command history feature, use the `no` form of this command.

```
terminal history

terminal no history
```

**Syntax Description**

This command has no arguments or keywords.

**Defaults**

Enabled, history buffer of 10 lines

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The history feature provides a record of commands you have entered. This feature is particularly useful for recalling long or complex commands or entries for the purposes of modifying them slightly and reexecuting them.

The `terminal history` command enables the command history feature with the default buffer size or the last buffer size specified using the `terminal history size` command.

Table 6 lists the keys and functions you can use to recall commands from the history buffer.

**Table 6  History Keys**

<table>
<thead>
<tr>
<th>Key(s)</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl-P or Up Arrow(^1)</td>
<td>Recalls commands in the history buffer in a backward sequence, beginning with the most recent command. Repeat the key sequence to recall successively older commands.</td>
</tr>
<tr>
<td>Ctrl-N or Down Arrow(^1)</td>
<td>Returns to more recent commands in the history buffer after recalling commands with Ctrl-P or the Up Arrow. Repeat the key sequence to recall successively more recent commands.</td>
</tr>
</tbody>
</table>

1. The arrow keys function only with ANSI-compatible terminals.

**Examples**

In the following example, the command history feature is disabled for the current terminal session:

```
Router> terminal no history
```
<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>history</td>
<td>Enables the command history function, or changes the command history buffer size for a particular line.</td>
</tr>
<tr>
<td></td>
<td>show history</td>
<td>Lists the commands you have entered in the current EXEC session.</td>
</tr>
<tr>
<td></td>
<td>terminal history size</td>
<td>Sets the size of the history buffer for the command history feature for the current terminal session.</td>
</tr>
</tbody>
</table>
**terminal history size**

To change the size of the command history buffer for the current terminal session, use the `terminal history size` EXEC mode command. To reset the command history buffer to its default size of 10 lines, use the `no` form of this command.

```
terminal history size number-of-lines
```

```
terminal no history size
```

**Syntax Description**

<table>
<thead>
<tr>
<th>number-of-lines</th>
<th>Number of command lines that the system will record in its history buffer. The range is from 0 to 256. The default is 10.</th>
</tr>
</thead>
</table>

**Defaults**

10 lines of command history

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The history feature provides a record of commands you have entered. This feature is particularly useful for recalling long or complex commands or entries for the purposes of modifying them slightly and reissuing them.

The `terminal history size` command enables the command history feature and sets the command history buffer size. The `terminal no history size` command resets the buffer size to the default of 10 command lines.

Table 6 lists the keys and functions you can use to recall commands from the history buffer. When you use these keys, the commands recalled will be from EXEC mode if you are in EXEC mode, or from all configuration modes if you are in any configuration mode.

**Table 7 History Keys**

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl-P or Up Arrow(^1)</td>
<td>Recalls commands in the history buffer in a backward sequence, beginning with the most recent command. Repeat the key sequence to recall successively older commands.</td>
</tr>
<tr>
<td>Ctrl-N or Down Arrow(^1)</td>
<td>Returns to more recent commands in the history buffer after recalling commands with Ctrl-P or the Up Arrow. Repeat the key sequence to recall successively more recent commands.</td>
</tr>
</tbody>
</table>

\(^1\) The arrow keys function only with ANSI-compatible terminals.
In EXEC mode, you can also use the `show history` command to show the contents of the command history buffer.

To check the current settings for the command history feature on your line, use the `show line` command.

**Examples**

In the following example, the number of command lines recorded is set to 15 for the current terminal session. The user then checks to see what line he/she is connected to using the `show users` command. The user uses this line information to issue the show line command. (In this example, the user uses the `show begin` option in the `show line` command to start the output at the “Editing is enabled/disabled” line.)

```
Router# terminal history size 15
Router# show users

  Line   User    Host(s)      Idle       Location
      * 50 vty 0  admin    idle     00:00:00
                ! the * symbol indicates the active terminal session for the user (line 50)

Router# show line 50 | begin Editing

Editing is enabled.
  ! the following line shows the history settings for the line
History is enabled, history size is 15.
DNS resolution in show commands is enabled
Full user help is disabled
Allowed transports are telnet. Preferred is none.
No output characters are padded
No special data dispatching characters
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>history</code></td>
<td>Enables the command history function, or changes the command history buffer size for a particular line.</td>
</tr>
<tr>
<td><code>show begin</code></td>
<td>Searches the output of any show command and displays the output from the first instance of a specified string.</td>
</tr>
<tr>
<td><code>show history</code></td>
<td>Lists the commands you have entered in the current EXEC session.</td>
</tr>
<tr>
<td><code>terminal history</code></td>
<td>Enables the command history feature for the current terminal session.</td>
</tr>
</tbody>
</table>