

# Chapter 5

## Command-Line Interface Operational Mode

When you log into the router and the CLI starts, you are at the top level of operational mode. At this level, there are a number of broad groups of CLI commands:

Commands for controlling the CLI environment—The commands in the set hierarchy configure the CLI display screen. For information about these commands, see “Control the CLI Environment” on page 73.

Commands for monitoring and troubleshooting—The following commands let you display information and statistics about the software and test network connectivity. Using these commands is discussed in the *JUNOS Internet Software Command Reference*.

clear—Clear statistics and protocol database information.

monitor—Perform real-time debugging of various software components, including the routing protocols and interfaces.

ping—Determine the reachability of a remote network host.

show—Display the current configuration and information about interfaces, routing protocols, routing tables, routing policy filters, and the chassis.

test —Test the configuration and application of policy filters and AS path regular expressions.

traceroute—Trace the route to a remote network host.

Commands for connecting to other network systems—The ssh command opens secure shell connections and the telnet command opens Telnet sessions to other hosts on the network. For information about these commands, see the *JUNOS Internet Software Command Reference*.

Commands for copying files—The file and copy commands copy files from one location on the router to another, from the router to a remote system, or from a remote system to the router. For information about these commands, see the *JUNOS Internet Software Command Reference*.

Commands for restarting software processes—The commands in the restart hierarchy restart the various JUNOS software processes, including the routing protocol, interface, and SNMP. For information about these commands, see the *JUNOS Internet Software Command Reference*.

A command—request—for performing system-level operations, including stopping and rebooting the router and loading JUNOS software images. For information about this command, see the *JUNOS Internet Software Command Reference*.

A command—start—to exit the CLI and start a UNIX shell. For information about this command, see the *JUNOS Internet Software Command Reference*.

A command—configure—for entering configuration mode, which provides a series of commands that configure the JUNOS software, including the routing protocols, interfaces, network management, and user access. For information about the CLI configuration commands, see “Configure the Router with the CLI” on page 77.

A command—quit—to exit the CLI. For information about this command, see the *JUNOS Internet Software Command Reference*.

For more information about the CLI operational mode commands, see the *JUNOS Internet Software Command Reference*.

This chapter discusses the following topics about the CLI:

Use the CLI on page 56

Set the Current Date and Time on page 71

Display CLI Command History on page 71

Monitor Who Uses the CLI on page 71

## Use the CLI

This section describes how to use the JUNOS software CLI. It discusses the following topics:

Get Help About Commands on page 57

Have the CLI Complete Commands on page 58

CLI Messages on page 59

Move around and Edit the Command Line on page 60

How Output Appears on the Screen on page 61

## Get Help About Commands

The CLI provides context-sensitive help at every level of the command hierarchy. The help information tells you which commands are available at the current level in the hierarchy and provides a brief description of each.

To get help while in the CLI, type ?. You do not need to press Enter after typing the question mark.

If you type the question mark at the command-line prompt, the CLI lists the available commands and options.

If you type the question mark after entering the complete name of a command or command option, the CLI lists the available commands and options, then redisplay the command names and options that you typed.

If you type the question mark in the middle of a command name, the CLI lists possible command completions that match the letters you have entered so far, then redisplay the letters that you typed.

### Examples: Get Help About Commands

List all available commands at the top level of the CLI's operational mode:

```
user@host> ?
Possible completions:
clear          Clear information in the system
configure     Manipulate software configuration information
file          Perform file operations
help          Provide help information
monitor       Real-time debugging
ping          Ping a remote target
quit          Exit the management session
request       Make system-level requests
restart       Restart a software process
set           Set CLI properties, date, time, craft display text
show         Show information about the system
ssh          Open a secure shell to another host
start        Start a software process
telnet       Telnet to another host
test         Diagnostic debugging commands
traceroute   Trace the route to a remote host
user@host>
```

List all commands that start with the letter c:

```
user@host> c?
Possible completions:
clear          Clear information in the system
configure     Manipulate software configuration information
user@host> c
```

List all available clear commands:

```

user@host> clear ?
Possible completions:
arp          Clear address-resolution information
bgp          Clear BGP information
chassis      Clear chassis information
firewall     Clear firewall counters
igmp         Clear IGMP information
interfaces  Clear interface information
isis         Clear IS-IS information
ldp          Clear LDP information
log          Clear contents of a log file
mpls         Clear MPLS information
msdp         Clear MSDP information
multicast    Clear Multicast information
ospf         Clear OSPF information
pim          Clear PIM information
rip          Clear RIP information
route        Clear routing table information
rsvp         Clear RSVP information
snmp         Clear SNMP information
system       Clear system status
vrrp         Clear VRRP statistics information
user@host> clear

```

## ***Have the CLI Complete Commands***

You do not always have to remember or type the full command or option name for the CLI to recognize it. To display all possible command or option completions, type the partial command followed immediately by a question mark.

To complete a command or option that you have partially typed, press a tab or a space. If the partially typed letters begin a string that uniquely identifies a command, the complete command name appears. Otherwise, a beep indicates that you have entered an ambiguous command, then displays the possible completions.

Command completion also applies to other strings, such as filenames and usernames. To display all possible values, type a partial string followed immediately by a question mark. However, to complete these strings, press a tab; pressing a space does not work.

**Examples: Use CLI Command Completion**

Issue the show interface command:

```
user@host> sh<Space>ow i<Space>
'i' is ambiguous.
Possible completions:
  igmp      Show information about IGMP
  interface Show interface information
  isis      Show information about IS-IS
user@host> show in<Space>terfaces <Enter>
Physical interface: at-0/1/0, Enabled, Physical link is Up
Interface index: 11, SNMP ifIndex: 65
Link-level type: ATM-PVC, MTU: 4482, Clocking: Internal, SONET mode
Speed: OC12, Loopback: None, Payload scrambler: Enabled
Device flags   : Present Running
Link flags     : 0x01
...
user@host>
```

Display a list of all log files whose name starts with the string “messages,” and then display the contents of one of the files:

```
user@myhost> show log mes?
Possible completions:
<filename>      Log file to display
messages        Size: 1417052, Last changed: Mar  3 00:33
messages.0.gz   Size: 145575, Last changed: Mar  3 00:00
messages.1.gz   Size: 134253, Last changed: Mar  2 23:00
messages.10.gz  Size: 137022, Last changed: Mar  2 14:00
messages.2.gr   Size: 137112, Last changed: Mar  2 22:00
messages.3.gz   Size: 121633, Last changed: Mar  2 21:00
messages.4.gz   Size: 135715, Last changed: Mar  2 20:00
messages.5.gz   Size: 137504, Last changed: Mar  2 19:00
messages.6.gz   Size: 134591, Last changed: Mar  2 18:00
messages.7.gz   Size: 132670, Last changed: Mar  2 17:00
messages.8.gz   Size: 136596, Last changed: Mar  2 16:00
messages.9.gz   Size: 136210, Last changed: Mar  2 15:00
user@myhost> show log mes<Tab>sages.4<Tab>.gz<Enter>
Jan 15 21:00:00 myhost newsyslog[1381]: logfile turned over
...
```

**CLI Messages**

You see messages when you enter and exit from configuration mode, when you commit a configuration, and when you type a string or value that is not valid.

When you commit a configuration, the JUNOS software checks the configuration you are committing. If there are no problems, a message indicates that the configuration was accepted. If there are problems, a message indicates where the errors are.

In the top-level CLI commands and in configuration mode, if you type an invalid string—for example, if you type the name of a command or statement that does not exist—you see the message “syntax error” or “unknown command.” A caret (^) indicates where the error is. Examples:

```

user@host> clear route
                    ^
syntax error, expecting <command>.

[edit]
user@host# telnet
                ^
unknown command.
    
```

When the number of choices is limited, a message might display the commands you can enter to correct the syntax error. For example,

```

[edit]
user@host# load myconfig-file<Enter>
                    ^
syntax error, expecting 'merge', 'override', or 'replace'.
    
```

### Move around and Edit the Command Line

In the CLI, you can use keyboard sequences to move around on a command line and edit the command line. You can also use keyboard sequences to scroll through a list of recently executed commands. Table 1 lists some of the CLI keyboard sequences. They are the same as those used in Emacs.

Table 1: CLI Keyboard Sequences

Category	Action	Keyboard Sequence
Move the Cursor	Move the cursor back one character.	Ctrl-b
	Move the cursor back one word.	Esc-b or Alt-b
	Move the cursor forward one character.	Ctrl-f
	Move the cursor forward one word.	Esc-f or Alt-f
	Move the cursor to the beginning of the command line.	Ctrl-a
	Move the cursor to the end of the command line.	Ctrl-e
Delete Characters	Delete the character before the cursor.	Ctrl-h, Delete, or Backspace
	Delete the character at the cursor.	Ctrl-d
	Delete all characters from the cursor to the end of the command line.	Ctrl-k
	Delete all characters on the command line.	Ctrl-u or Ctrl-x
	Delete the word before the cursor.	Ctrl-w, Esc-Backspace, or Alt-Backspace
	Delete the word after the cursor.	Esc-d or Alt-d
Insert Recently Deleted Text	Insert the most recently deleted text at the cursor.	Ctrl-y

Category	Action	Keyboard Sequence
Redraw the Screen	Redraw the current line.	Ctrl-l
Display Previous Command Lines	Scroll backward through the list of recently executed commands.	Ctrl-p
	Scroll forward through the list of recently executed commands.	Ctrl-n
	Search the CLI history in reverse order for lines matching the search string.	Ctrl-r
	Search the CLI history by typing some text at the prompt, followed by the keyboard sequence. The CLI attempts to expand the text into the most recent word in the history for which the text is a prefix.	Esc-/
Repeat Keyboard Sequences	Specify the number of times to execute a keyboard sequence. <i>number</i> can be from 1 through 9.	Esc- <i>number</i> <i>sequence</i> or Alt- <i>number</i> <i>sequence</i>

## How Output Appears on the Screen

When you issue commands in operational mode, or when you issue the show command in configuration mode, the output appears on the screen. You can also filter the output of commands, either to perform simple commands on the output or to place the output into a file.

### Display Output One Screen at a Time

If the output is longer than the screen length, it appears one screen at a time using a UNIX more-type interface. The prompt `--More--` indicates that more output is available. Table 2 lists the keyboard sequences you can use at the `--More--` prompt. As soon as the CLI knows how long the output is (usually by the second screen), it displays the percentage of the command output above the prompt.

Table 2: `--More--` Prompt Keyboard Sequences

Category	Action	Keyboard Sequence
Get Help	Display information about the keyboard sequences you can display at the <code>--More--</code> prompt.	h
Scroll Down	Scroll down one line.	Enter, Return, k, Ctrl-m, Ctrl-n, or down arrow
	Scroll down one-half screen.	Tab, d, Ctrl-d, or Ctrl-x
	Scroll down one whole screen.	Space or Ctrl-f
	Scroll down to the bottom of the output.	Ctrl-e or G
	Display the output all at once instead of one screen at a time. (Same as specifying the   no-more command.)	N
Scroll Up	Display the previous line of output.	j, Ctrl-h, Ctrl-p, or up arrow
	Scroll up one-half screen.	u or Ctrl-u
	Scroll up one whole screen.	b or Ctrl-b
	Scroll up to the top of the output.	Ctrl-a or g

Category	Action	Keyboard Sequence
Search	Search forward for a string.	/string
	Search backward for a string.	?string
	Repeat the previous search for a string.	n
	Search for a text string. You are prompted for the string to match. (Same as specifying the   match string command.)	m or M
	Search, ignoring a text string. You are prompted for the string to not match. (Same as specifying the   except string command.)	e or E
Interrupt or End Output, Redraw the Output, and Save the Output to a File	Interrupt the display of output.	Ctrl-C, q, Q, or Ctrl-k
	Do not redisplay the CLI prompt immediately after displaying the output, but remain at the --More-- prompt. (Same as specifying the   hold command.)	H
	Clear any match conditions and display the complete output.	c or C
	Redraw the output on the screen.	Ctrl-l
	Save the command output to a file. You are prompted for a filename. (Same as specifying the   save filename command.)	s or S

## Filter Command Output

For operational and configuration commands that display output, such as the show commands, you can filter the output. When you display help about these commands, one of the options listed is |, called a *pipe*, which allows you to filter the command output. For example:

```

user@host> show configuration ?
Possible completions:
  <[Enter]> Execute this command
  |         Pipe through a command
user@host> show configuration | ?
Possible completions:
count      Count occurrences
except     Show only text that does not match a pattern
find       Search for the first occurrence of a pattern
hold       Hold text without exiting the --(more)-- prompt
match      Show only text that matches a pattern
no-more    Don't paginate output
save       Save output text to a file

```

In configuration mode, two additional filters appear; display and compare:

```

[edit]
user@host # show | ?
Possible completions:
compare    Compare configuration changes with a prior version
count      Count occurrences
display    Display additional configuration information
except     Show only text that does not match a pattern
find       Search for the first occurrence of a pattern
hold       Hold text without exiting the ---More--- prompt
match      Show only text that matches a pattern
no-more    Don't paginate output
save       Save output text to a file

```

The following filtering operations are available:

- Place Command Output in a File on page 63
- Search for a String in the Output on page 63
- Compare Configuration Changes with a Prior Version on page 66
- Count the Number of Lines in the Output on page 68
- Display All Output at Once on page 68
- Retain the Output after the Last Screen on page 68
- Display Additional Information about the Configuration on page 68
- Filter Command Output Multiple Times on page 70

### *Place Command Output in a File*

When the output is very long, when you need to store or analyze the output, or when you need to email or FTP the output, you can place the output of a command into a file. Doing this is useful when the output scrolls off the screen, making it difficult to cut the output from a window and paste it into another.

To save the output to a file, specify the `save` command after the pipe:

```
user@host> command | save filename
```

By default, the file is placed in your home directory on the router. For information about how you can specify the name of the file, see “How to Specify Filenames” on page 184.

This example stores the output of the `request support information` command in a file:

```
user@host> request support information | save filename
Wrote 1143 lines of output to 'filename'
user@host>
```

### *Search for a String in the Output*

You can filter the output to search for a text matching a regular expression. You can match a regular expression match everything except a regular expression, or find the first occurrence of text matching a regular expression. All searches are not case-sensitive.

To match a regular expression, specify the `match` command after the pipe:

```
user@host> command | match regular-expression
```

To ignore text that matches a regular expression, specify the `except` command after the pipe:

```
user@host> command | except regular-expression
```

If the *regular-expression* contains any spaces, operators, or wildcard characters, enclose it in quotation marks.

You use extended regular expressions to specify what text in the output to match. Command regular expressions implement the extended (modern) regular expressions as defined in POSIX 1003.2. Table 3 lists common regular expression operators.

**Table 3: Common Regular Expression Operators**

Operator	Match...
	One of the two terms on either side of the pipe.
^	At the beginning of an expression, used to denote where the command begins, where there might be some ambiguity.
\$	Character at the end of a command. Used to denote a command that must be matched exactly up to that point. For example, allow-commands "show interfaces \$" means that the user cannot issue show interfaces detail or show interfaces extensive.
[ ]	Range of letters or digits. To separate the start and end of a range, use a hyphen ( - ).
( )	A group of commands, indicating an expression to be evaluated and the result is then evaluated as part of the overall expression.

For example, if a command produces the following output:

```
one two
two two
three two one
four
```

The match two command displays:

```
one two
two two
three two one
```

The except one command displays:

```
two two
four
```

List all the ATM interfaces in the configuration:

```
user@host> show configuration | match at-
at-2/1/0 {
at-2/1/1 {
at-2/2/0 {
at-5/2/0 {
at-5/3/0 {
```

Display a skeleton of your router configuration:

```
[edit]
user@host # show | match {
system {
  root-authentication {
  name-server {
  login {
    class superuser {
    user junipero {
      authentication {
  services {
  syslog {
    file messages {
  processes {
chassis {
  alarm {
    sonet {
  images {
    scb {
    fpc {
  interfaces {
    at-2/1/1 {
      atm-options {
      unit 0 {
    at-2/2/0 {
    ...
snmp {
  community public {
    clients {
  routing-options {
    static {
      route 0.0.0.0/0 {
      route 192.168.0.0/16 {
      route 208.197.169.0/24 {
  protocols {
    rsvp {
      interface so-5/1/0 {
    mpls {
      interface so-5/1/0 {
    bgp {
      group internal {
    ospf {
      area 0.0.0.0 {
      interface so-5/1/0 {
```

List all users who are logged into the router except for the user “root”:

```
user@host> show system users | except root
 8:28PM up 1 day, 13:59, 2 users, load averages: 0.01, 0.01, 0.00
USER   TY  FROM          LOGIN@  IDLE WHAT
sheep  p0  baa.juniper.net 7:25PM  - cli
```

Save the configuration, except for encrypted passwords, to a file:

```
user@host> show configuration | except SECRET-DATA | save my.output.file
```

Display the output, starting not at the beginning but rather at the first occurrence of text matching a regular expression, using the `find` command after the pipe:

```
user@host> command | find regular-expression
```

If the regular expression contains spaces, operators, or wildcard characters, enclose the expression in quotation marks.

List the routes in the routing table starting at 208.197.169.0:

```
user@host> show route | find 208.197.169.0
208.197.169.0/24    *[Static/5] 1d 13:22:11
                  > to 192.168.4.254 via so-3/0/0.0
224.0.0.5/32      *[OSPF/10] 1d 13:22:12, metric 1

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

47.0005.80ff.f800.0000.0108.0001.1921.6800.4015.00/160
                  *[Direct/0] 1d 13:22:12
                  > via lo0.0
```

### Compare Configuration Changes with a Prior Version

In configuration mode only, when you have made changes to the configuration and want to compare the candidate configuration with a prior version, you can use the `compare` command to display the configuration. The `compare` command compares the candidate configuration with either the current committed configuration or a configuration file and displays the differences between the two configurations. To compare configurations, specify the `compare` command after the pipe:

```
[edit]
user@host# show | compare [filename | rollback n]
```

*filename* is the full path to a configuration file. The file must be in the proper format, a hierarchy of statements. For information about how to save a configuration to a file, see “Save a Configuration to a File” on page 103. For information about formatting the hierarchy of statements, see “Configuration Statement Hierarchy” on page 78.

*n* is the index into the list of previously committed configurations. The most recently saved configuration is number 0, and the oldest saved configuration is number 9. If you do not specify arguments, the candidate configuration is compared against the active configuration file (`/config/juniper.conf`).

The comparison output uses the following conventions:

Statements that are only in the candidate configuration are prefixed with a plus sign (+).

Statements that are only in the comparison file are prefixed with a minus sign (-).

Statements that are unchanged are prefixed with a single blank space ( ).

The following example shows various changes, then a comparison of the candidate configuration with the active configuration, showing only the changes made at the [edit protocols bgp] hierarchy level.

```
[edit]
user@host# edit protocols bgp

[edit protocols bgp]
user@host# show
group "my group" {
  type internal;
  hold-time 60;
  advertise-inactive;
  allow 1.1.1.1/32;
}
group fred {
  allow 2.2.2.2/32;
}
group "test peers" {
  type external;
  allow 3.3.3.3/32;
}
[edit protocols bgp]
user@host# set group "my group" hold-time 90
[edit protocols bgp]
user@host# delete group "my group" advertise-inactive
[edit protocols bgp]
user@host# set group fred advertise-inactive
[edit protocols bgp]
user@host# delete group "test peers"
[edit protocols bgp]
user@host# show | compare
group "my group" {
  type internal;
- hold-time 60;
+ hold-time 90;
- advertise-inactive;
  allow 1.1.1.1/32;
}
group fred {
+ advertise-inactive;
  allow 2.2.2.2/32;
}
-group "test peers" {
- type external;
- allow 3.3.3.3/32;
-}
[edit protocols bgp]
user@host# show
group "my group" {
  type internal;
  hold-time 90;
  allow 1.1.1.1/32;
}
group fred {
  advertise-inactive;
  allow 2.2.2.2/32;
}
```

To show only the changes between the two configurations, use a match command:

```
user@host# show | compare | match "^[+-]"
```

### Count the Number of Lines in the Output

To count the number of lines in the output, specify the count command after the pipe:

```
user@host> command | count
```

For example:

```
user@host> show configuration | count
Count: 269 lines
user@host> show route | count
Count: 67 lines
```

### Display All Output at Once

To display the output all at once instead of one screen at a time, specify the no-more command after the pipe. This command is equivalent to the set cli screen-length 0 command, but affects the output of the one command only.

```
user@host> command | no-more
```

### Retain the Output after the Last Screen

When you view output one screen at a time, you typically return to the CLI prompt after viewing the last screen.

To not return immediately, use the hold command after the pipe. This feature is useful, for example, when you want to scroll or search through the output.

```
user@host> command | hold
```

### Display Additional Information about the Configuration

In configuration mode only, to display additional information about the configuration, use the display detail command after the pipe in conjunction with a show command. The additional information includes the help string that explains each configuration statement and the permission bits required to add and modify the configuration statement.

```
user@host> show <hierarchy-level> | display detail
```

For example:

```
[edit]
user@host> show | display detail
##
## version: Software version information
## require: system
##
version "3.4R1 [tlim]";
system {
##
## host-name: Host name for this router
## match: ^[:alnum:]._]+$
## require: system
##
host-name router-name;
```

```

##
## domain-name: Domain name for this router
## match: ^[[:alnum:]]+$
## require: system
##
domain-name isp.net;
##
## backup-router: Address of router to use while booting
##
backup-router 192.168.100.1;
root-authentication {
    ##
    ## encrypted-password: Crypted password string
    ##
    encrypted-password "$1$BYJQE$/ocQof8pmcm7MSGK0"; # SECRET-DATA
}
##
## name-server: DNS name servers
## require: system
##
name-server {
    ##
    ## name-server: DNS name server address
    ##
    208.197.1.0;
}
login {
    ##
    ## class: User name (login)
    ## match: ^[[:alnum:]]+$
    ##
    class superuser {
        ##
        ## permissions: Set of permitted operation categories
        ##
        permissions all;
    }
    ...
}
##
## services: System services
## require: system
##
services {
    ## services: Service name
    ##
    ftp;
    ##
    ## services: Service name
    ##
    telnet;
    ##
}
syslog {
    ##
    ## file-name: File to record logging data
    ##
    file messages {
        ##
        ## Facility type
        ## Level name
        ##
        any notice;
        ##
    }
}

```

```

    ## Facility type
    ## Level name
    ##
    authorization info;
  }
}
chassis {
  alarm {
    sonet {
      ##
      ## lol: Loss of light
      ## alias: loss-of-light
      ##
      lol red;
    }
  }
}
interfaces {
  ##
  ## Interface name
  ##
  at-2/1/1 {
    atm-options {
      ##
      ## vpi: Virtual path index
      ## range: 0 .. 255
      ## maximum-vcs: Maximum number of virtual circuits on this VP
      ##
      vpi 0 maximum-vcs 512;
    }
    ##
    ## unit: Logical unit number
    ## range: 0 .. 16384
    ##
    unit 0 {
      ##
      ## vci: ATM point-to-point virtual circuit identifier ([vpi].[vci])
      ## match: ^([[:digit:]]+)(0,1)[[:digit:]]+$
      ##
      vci 0.128;
    }
  }
}
...

```

### Filter Command Output Multiple Times

For the output of a single command, you can filter the output one or more times. For example:

```

user@host> command | match regular-expression | except regular-expression | match
other-regular-expression | find regular-expression | hold

```

## Set the Current Date and Time

To set the current date and time on the router, use the `set date` command:

```
user@host> set date YYYYMMDDhhmm.ss
```

YYYY is the four-digit year, MM is the two-digit month, DD is the two digit date, hh is the two-digit hour, mm is the two-digit minute, and ss is the two-digit second. At a minimum, you must specify the two-digit minute. All other parts of the date and time are optional.

To set the time zone, see “Set the Time Zone” on page 213. To configure time synchronization, see “Configure the Network Time Protocol” on page 214.

## Display CLI Command History

You can display a list of recent commands that you issued. To display the command history, use the `show cli history` command:

```
user@host> show cli history
03-03 01:00:50 -- show cli history
03-03 01:01:12 -- show interfaces terse
03-03 01:01:22 -- show interfaces lo0
03-03 01:01:44 -- show bgp next-hop-database
03-03 01:01:51 -- show cli history
```

By default, this command displays the last 100 commands issued in the CLI. If you specify a number with the command, it displays that number of recent commands. For example:

```
user@host> show cli history 3
01:01:44 -- show bgp next-hop-database
01:01:51 -- show cli history
01:02:51 -- show cli history 3
```

## Monitor Who Uses the CLI

Depending upon how you configure the JUNOS software, multiple users can log in to the router, use the CLI, and configure or modify the software configuration.

The JUNOS software provides a general syslog-like mechanism to log system operations, such as when users log in to the router and when they issue CLI commands. To configure system logging, include the `syslog` statement in the configuration, as described in the section “Configure System Logging” on page 219.

If, when you enter configuration mode, another user is also in configuration mode, a notification message is displayed that indicates who the user is and what portion of the configuration they are viewing or editing:

```
user@host> configure
Entering configuration mode
Current configuration users:
  root terminal p3 (pid 1088) on since 1999-05-13 01:03:27 EDT
  [edit interfaces so-3/0/0 unit 0 family inet]
The configuration has been changed but not committed
```

