Chapter 2

Cheat Sheet for the CLI Commands

This chapter provides quick reference information for the JUNOS software command-line interface (CLI). For more detailed information about using the CLI, see “Command-Line Interface Overview” on page 321.

CLI Operational Mode Top-Level Commands

In operational mode, you enter commands to monitor and diagnose the software, network connectivity, and the router. When you log in to the router and the CLI starts, you are at the top level of the CLI operational mode. At this level, there are several broad groups of CLI commands. Table 10 lists the top-level CLI operational mode commands and describes the options available for each command. The commands are listed in alphabetical order.

Table 10: CLI Operational Mode Top-Level Commands

Command	Description
clear	Clear statistics and protocol database information. Syntax: clear (arp | bgp | firewall | helper | igmp | ike | ilmi | interfaces | ipsec | ipv6 | isis | ldp | log | mpls | msdp | multicast | ospf | pim | rip | ripng | route | rsvp | snmp | system | vrrp)
configure	Enter CLI configuration mode. Alternative commands: configure <exclusive> <private>
file	Perform file manipulation operations, such as copy, delete, list, rename, and show. Syntax: file (compare | copy | delete | list | rename | show)
help	Provide help information. Syntax: help (reference | syslog | topic)
monitor	Monitor a log file or interface traffic in real time. Syntax: monitor (interface | list | start | stop | traffic)
mtrace	Display trace information about a multicast path from a source to a receiver. Syntax: mtrace (from-source | monitor | to-gateway)
ping	Verify IP connectivity to another IP host or Asynchronous Transfer Mode (ATM) connectivity (ping ATM) using Operation Administration and Maintenance (OAM) cells to an ATM endstation. Syntax: ping *host* <interface *source-interface*> <bypass-routing> <count *requests*> <do-not-fragment> <interval *seconds*> <pattern *string*> <record-route> <routing-instance *routing-instance-name*> <size *bytes*> <strict> <tos *type-of-service*> <ttl *value*> <via *route*> <rapid | detail> Syntax: ping atm interface *interface* <count *count*> <end-to-end | segment> <interval *interval*> <sequence-number *sequence-number*> <vci *vci*> <brief> Syntax: ping vpn-interface *vpn-interface host* <local *echo-address*>
pipe	Filter the output of an operational mode or configuration mode command. Syntax: | (compare | count | display <detail | inheritance | xml> | except *pattern* | find *pattern* | last *lines* | match *pattern* | no-more | resolve <file-names> | save *filename* | trim *columns)*
quit	Log out from the CLI process. Syntax: quit
request	Make system-level requests, such as halt or reboot the router, load software packages, and back up the router's file systems. Syntax: request system (halt | reboot | snapshot | software)
restart	Restart the router hardware or software processes. Syntax: restart (fpc | class-of-service | gracefully | immediately | interface-control | mib-process | network-access-service | remote-operations | routing | sampling | sfm | snmp | soft)
set	Set CLI properties, the router's date and time, and the craft interface display text. Syntax: set (chassis | cli | date)

Command	Description
show	Show information about all aspects of the software, including interfaces and routing protocols. Syntax: show (accounting | aps | arp | as-path | bgp | chassis | cli | configuration | connections | dvmrp | firewall | helper | host | igmp | ike | ilmi | interfaces | ipsec | ipv6 | isis | l2circuit | l2vpn | ldp | link-management | log | mpls | msdp | multicast | ntp | ospf | pfe | pim | policer | policy | rip | ripng | route | rsvp | sap | snmp | system | task | ted | version | vrrp)
ssh	Open a secure shell to another host. Syntax: ssh *host* <bypass-routing> <routing-instance *routing-instance-name*> <source *address*> <vpn-interface *vpn-interface*> <v1 | v2>
start	Start a software process. Syntax: start shell
telnet	Start a telnet session to another host. Syntax: telnet *host* <8bit> <bypass-routing> <inet | inet6> <noresolve> <port *port*> <interface *interface-name*> <routing-instance *routing-instance-name*> <source *address*> <vpn-interface *vpn-interface*>
test	Run various diagnostic debugging commands. Syntax: test (configuration | interface | msdp | policy)
traceroute	Trace the route to a remote host. Syntax: traceroute *host* <as-number-lookup> <bypass-routing> <gateway *address*> <inet | inet6> <noresolve> <routing-instance *routing-instance-name*> <*source address*> <tos *value*> <ttl *value*> <vpn-interface *vpn-interface*> <wait *seconds*>

CLI Configuration Mode Top-Level Commands

In configuration mode, you configure the JUNOS software by creating a hierarchy of configuration statements. You can do this using the CLI or by creating a text (ASCII) file that contains the statement hierarchy. (The statement hierarchy is identical in both the CLI and the text configuration file.) You can configure all properties of the JUNOS software, including interfaces, general routing information, routing protocols, and user access, as well as several system hardware properties. When you have finished entering the configuration statements, you commit them, which activates the configuration on the router.

Table 11 lists each CLI configuration mode command and describes the options available for each command. The commands are organized alphabetically.

Table 11: CLI Configuration Mode Commands

Command	Description
activate	Remove the **inactive:** tag from a statement, effectively reading the statement or identifier to the configuration. Statements or identifiers that have been activated take effect when you next issue the **commit** command. Syntax: **activate** (*statement-path* | *identifier*)
annotate	Add comments to a configuration. Syntax: **annotate** <*statement-path*> "*comment-string*"
commit	Commit the set of changes to the database and cause the changes to take operational effect. Syntax: **commit** <and-quit> <check> <confirmed <*minutes*>> <synchronize>
copy	Make a copy of an existing statement in the configuration. Syntax: **copy** <*statement-path*> *identifier 1* **to** *identifier 2*
deactivate	Add the **inactive:** tag to a statement, effectively commenting out the statement or identifier from the configuration. Statements or identifiers marked as inactive do not take effect when you issue the **commit** command. Syntax: **deactivate** (*statement-path* | *identifier*)
delete	Delete a statement or identifier. All subordinate statements and identifiers contained within the specified statement path are deleted with it. Syntax: **delete** (*statement-path* | *identifier*)
edit	Move inside the specified statement hierarchy. If the statement does not exist, it is created. Syntax: **edit** <*statement-path*>
exit	Exit the current level of the statement hierarchy, returning to the level prior to the last **edit** command, or exit from configuration mode. The **quit** and **exit** commands are synonyms. Syntax: **exit** <configuration-mode>
help	Display help about available configuration statements. Syntax: **help** (apropos | reference | syslog | topic) <*string*>
insert	Insert an identifier into an existing hierarchy. Syntax: **insert** <*statement-path*> *identifier1* (before | after) *identifier2*
load	Load a configuration from an ASCII configuration file or from terminal input. Your current location in the configuration hierarchy is ignored when the load operation occurs. Syntax: **load** (merge | override | replace) (*filename* | terminal)

Command	Description
quit	Exit the current level of the statement hierarchy, returning to the level prior to the last **edit** command, or exit from configuration mode. The **quit** and **exit** commands are synonyms. Syntax: **quit** <configuration-mode>
rename	Rename an existing configuration statement or identifier. Syntax: **rename** <*statement-path*> *identifier1* **to** *identifier2*
rollback	Return to a previously committed configuration. The software saves the last 10 committed configurations, including the rollback number, date, time, and name of the user who issued the commit configuration command. **rollback 0** erases any configuration changes made to the current candidate configuration. The currently operational JUNOS software configuration is stored in the file **juniper.conf**, and the last three committed configurations are stored in the files **juniper.conf.1.gz**, **juniper.conf.2.gz**, and **juniper.conf.3.gz**. These four files are located in the directory **/config/**, which is on the router's flash drive. The remaining six previous committed configurations, the files **juniper.conf.4.gz** through **juniper.conf.9.gz,** are stored in the directory **/var/db/config/**, which is on the router's hard disk. Syntax: **rollback** <*number*>
run	Run an operational mode CLI command without exiting from configuration mode. Syntax: **run** <*operation-command*>
save	Save the configuration to an ASCII file in the user's home directory (by default) or to the user's terminal session. The statement hierarchy and the contents of the current level of the statement hierarchy (and below) are saved. This allows a section of the configuration to be saved, while fully specifying the statement hierarchy. Syntax: **save** *filename* | *terminal*
set	Create a statement hierarchy and set identifier values. This is similar to the **edit** command except that your current level in the hierarchy does not change, and you can set identifier values, while the **edit** command only allows access to a statement path. Syntax: **set** (*statement-path* | *identifier*)
show	Display the current configuration. Syntax: **show** (*statement-path* | *identifier*)
status	Display the users currently editing the configuration. Syntax: **status**
top	Return to the top level of configuration command mode, indicated by the **[edit]** banner, or execute a command from the top level of the configuration. Syntax: **top** <*configuration-command*>
up	Move up one level in the statement hierarchy. Syntax: **up** <*number*>
update	Update a private database. For more information on the **update** command, see the *JUNOS System Basics and Services Command Reference*. Syntax: **update**

Load a Configuration Using Copy and Paste Commands

You can load configurations using the copy and paste commands in the following ways:

1. Load a Configuration from a File To a Router on page 22
2. Load a Configuration Using the display set Command on page 24

Load a Configuration from a File To a Router

Purpose You can create a file, copy the file to the local router, and then load the file into the CLI. After you have loaded the file, you can commit it to activate the configuration on the router, or you can edit the configuration interactively using the CLI and commit it at a later time.

Action To load a configuration from a file, follow these steps:

1. Create the configuration in a file using a text editor such as Notepad, making sure that the syntax of the configuration file is correct. See *JUNOS Internet Software Protocols, Class of Service, Chassis, and Management Command Reference*, for information about testing the syntax of a configuration file.
2. In the text file, use an option to perform the required action. The following table lists and describes some options. For an example of a text file, see "What It Means."

Table 12: Options for the load Command

merge	Combines the current configuration and the configuration in *filename* or the one that you type at the terminal. A merge operation is useful when you are adding a new section to an existing configuration. If the existing configuration and the incoming configuration contain conflicting statements, the statements in the incoming configuration override those in the existing configuration.
override	Discards the current candidate configuration and loads the configuration in *filename* or the one that you type at the terminal. When you use the override option and commit the configuration, all system processes reparse the configuration. You can use the override option at any level of the hierarchy.
replace	Searches for the replace tags, deletes the existing statements of the same name, if any, and replaces them with the incoming configuration. If there is no existing statement of the same name, the replace operation adds the statements marked with the replace tag to the configuration. Note: For this operation to work, you must include replace tags in the text file or configuration you type at the terminal.

3. Enter Ctrl+a to select all the text, and Ctrl+c to copy the contents of the text file to the clipboard.

4. On the router, enter configuration mode:

   ```
   user@host> cli
   [edit]
   user@host#
   ```

5. Load the configuration file:

   ```
   user@host> load merge terminal
   ```

6. At the prompt, paste the contents of the clipboard using the mouse and the paste icon.

   ```
   [edit]
   user@host# load merge terminal
   [Type ^D at a new line to end input]
   > Paste the contents of the clipboard here<
   ```

7. Hit Enter.

8. Enter Ctrl+d.

9. Commit the configuration to activate it on the router, or you can edit the configuration interactively using the CLI and commit it at a later time.

Sample Output The following is an example of a text file with the replace option:

```
interfaces {
replace:
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.34.1/30;
            }
        }
    }

protocols {
replace:
    isis {
        interface so-0/0/1.0 {
            level 1 metric 10;
            level 2 disable;
        }
        interface fxp0.0 {
            disable;
        }
        interface lo0.0;
    }
}
```

The following output is for Step 4 through Step 8:

```
[edit]
user@R1# load merge terminal
[Type ^D at a new line to end input]
interfaces {
replace:
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.34.1/30;
            }
        }
    }
protocols {
replace:
    isis {
        interface so-0/0/1.0 {
            level 1 metric 10;
            level 2 disable;
        }
        interface fxp0.0 {
            disable;
        }
        interface lo0.0;
    }
}
load complete
```

What It Means The sample output shows a configuration loaded from a text file with the replace option. For more information about loading a configuration, see the *JUNOS System Basics Configuration Guide*.

Load a Configuration Using the display set Command

Purpose In configuration mode only, you can display the configuration as a series of configuration mode commands required to recreate the configuration. This is useful for users who are not familiar with how to use configuration mode commands or for users who wish to cut, paste, and edit the displayed configuration. In addition, you can duplicate the configuration of one router to another.

Action To load a configuration from the local router to a target router, follow these steps:

1. On the local router, enter configuration mode:

   user@R1> **cli**
   [edit]
   user@host#

2. Go to the hierarchy level you want to copy. For example:

   [edit]
   user@R1# **edit interfaces**

3. Display the series of configuration commands required to recreate the configuration. For example:

   ```
   [edit interfaces]
   user@R1# show | display set
   set interfaces so-0/0/0 unit 0 family inet accounting destination-class-usage
   set interfaces so-0/0/0 unit 0 family inet address 10.1.12.1/30
   set interfaces fxp0 unit 0 family inet address 10.168.70.143/21
   set interfaces lo0 unit 0 family inet address 10.0.0.1/32
   set interfaces lo0 unit 0 family iso address 49.0002.1000.0000.0003.00
   ```

4. Copy each line of the configuration individually from the local router to the target router. In the target router, you must be at the top level of the configuration and in configuration mode. For example:

   ```
   mwazna@R2> edit
   Entering configuration mode

   [edit]
   mwazna@R2# set interfaces so-0/0/0 unit 0 family inet accounting
   destination-class-usage
   ```

5. Continue cutting and pasting each line of the configuration.

6. Commit the configuration to activate it on the router, or you can edit the configuration interactively using the CLI and commit it at a later time.

CLI Keyboard Shortcuts

In the CLI, you can use keyboard sequences to move around and edit a command line. You can also use keyboard sequences to scroll through a list of recently executed commands.

The following table lists some of the CLI keyboard sequences.

Table 13: CLI Keyboard Shortcuts

Keyboard sequence	Action
Ctrl+b	Move the cursor back one character.
Esc+b or Alt+b	Move the cursor back one word.
Ctrl + f	Move the cursor forward one character.
Esc+f or Alt+f	Move the cursor forward one word.
Ctrl+a	Move the cursor to the beginning of the command line.
Ctrl+e	Move the cursor to the end of the command line.
Ctrl+h, Delete, or Backspace	Delete the character before the cursor.
Ctrl+d	Delete the character at the cursor.
Ctrl+k	Delete the all characters from the cursor to the end of the command line.
Ctrl+u or Ctrl+x	Delete the all characters from the command line.
Ctrl+w, Esc + Backspace, or Alt + Backspace	Delete the word before the cursor.
Esc+d or Alt+d	Delete the word after the cursor.
Ctrl+y	Insert the most recently deleted text at the cursor.
Ctrl+l	Redraw the current line.
Ctrl+p	Scroll backward through the list of recently executed commands.
Ctrl+n	Scroll forward through the list of recently executed commands.
Ctrl+r	Search the CLI history incrementally in reverse order for lines matching the search string.
Esc+/ or Alt+/	Search the CLI history for words for which the current word is a prefix.
Esc-1 through Esc-9 or Alt-1 through Alt-9	Specify the number of times to execute a keyboard sequence.

Manage Output at the —(more)— Prompt

If the output is longer than the screen length, it appears one screen at a time with the UNIX —(more)– prompt at the end of the screen. The —(more)— prompt indicates that more output is available. The following table lists the keyboard sequences you can use at the –(more)— prompt.

Table 14: Keyboard Shortcuts at the —(more)— Prompt

Keyboard Shortcut	Action
Enter, Return, k, Ctrl+m, Ctrl+n, or down arrow	Scroll down one line.
Tab, d, Ctrl+d, or Ctrl+x	Scroll down one-half screen.
Space or Ctrl+f	Scroll down one whole screen.
Ctrl+e or g	Scroll down to the bottom of the output.
n (or no-more)	Display the output all at once instead of one screen at a time.
j, Ctrl-h, Ctrl-p, or up arrow	Scroll up one line.
u or Ctrl-u	Scroll up one-half screen.
b or Ctrl-b	Scroll up one whole screen.
Ctrl-a or g	Scroll up to the bottom of the output.
/*string*	Search forward for a string.
?*string*	Search backward for a string.
n	Repeat previous search for a string.
m or M (or | match *string*)	Find a text string. You are prompted for the string to match
e or E (or | except *string*)	Find, ignoring a text string. You are prompted for the string to ignore.
Ctrl-C, q, Q, or Ctrl-k	Interrupt the display of output.
H (Same as specifying | hold)	Hold the CLI at the More prompt after displaying all output.
c or C	Clear any match conditions and display the complete output.
Ctrl-l	Redraw the output on the screen.
s or S (or | save *filename*)	Save the command output to a file. You are prompted for a filename.