

Appendix A

Command-Line Interface Overview

This appendix provides an overview of the JUNOS software command-line interface (CLI). For more detailed information about using the JUNOS software CLI, see the *JUNOS Interfaces Command Reference*, *JUNOS Routing Protocols Command Reference*, and the *System Basics and Services Command Reference*.

The CLI is the interface to the software that you use whenever you access the router—whether from the console or through a remote network connection. The CLI, which automatically starts after the router finishes booting, provides commands that you use to perform various tasks, including configuring the JUNOS software, and monitoring and troubleshooting the software, network connectivity, and the router hardware.

The CLI has two modes:

- CLI Operational Mode on page 166
- CLI Configuration Mode on page 172

CLI Operational Mode

In operational mode you enter commands to monitor and troubleshoot the software, network connectivity, and router by entering commands.

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 (see Table 23).

Table 23: CLI Operational Mode Top-Level Commands

Command	Description
clear	Clear statistics and protocol database information. Syntax: clear <arp bfd bgp firewall helper igmp ike ilmi interfaces ipv6 isis ldp log mld mpls msdp multicast ospf pim rip ripng route rsvp services 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 <archive source destination compare compress source destination 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> <start stop list> <traffic>
mtrace	Display trace information about a multicast path from a source to a receiver. Syntax: mtrace <source-name> <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 <i>host-name</i> <interface <i>source-interface</i> > <bypass-routing> <count requests> <detail> <do-not-fragment> <inet inet6> <interval seconds> <logical-router <i>logical-router-name</i> > <loose-source value> <pattern string> <rapid> <record-route> <routing-instance <i>routing-instance-name</i> > <size bytes> <strict strict-source value> <tos type-of-service> <ttl value> <verbose> <via route> <wait seconds> Syntax: ping atm interface <i>interface</i> <count <i>count</i> > <end-to-end segment> <interval <i>interval</i> > <sequence-number <i>sequence-number</i> > <vci <i>vci</i> > <brief> Syntax: ping vpn-interface <i>vpn-interface</i> <i>host</i> <local <i>echo-address</i> >
pipe	Filter the output of an operational mode or configuration mode command. Syntax: <compare <filename rollback n> count display <detail inheritance xml> except pattern find pattern hold last match pattern no-more resolve <full-names> save filename trim columns>
quit	Log out from the CLI process. Syntax: quit

Command	Description
request	<ul style="list-style-type: none"> ■ Stop or reboot router components, switch between primary and backup components, display messages, and display system information. <p>Syntax: <code>request <chassis ipsec switch message routing-engine security services flow-collector support information></code></p> <ul style="list-style-type: none"> ■ Stop or reboot the router, load software packages, and back up the router's file systems. <p>Syntax: <code>request <chassis ipsec switch message routing-engine security services></code></p>
restart	<p>Restart the router hardware or software processes.</p> <p>Syntax: <code>restart <adaptive-services chassis-control class-of-service disk-monitoring ecc-error-logging firewall interface-control kernel-replication l2tp-service mib-process network-access-service pgm pic-services-logging remote-operations routing sampling snmp> <gracefully immediately soft></code></p>
set	<p>Set CLI properties, the router's date and time, and the craft interface display text.</p> <p>Syntax: <code>set <chassis cli date date ntp></code></p>
show	<p>Show information about all aspects of the software, including interfaces and routing protocols.</p> <p>Syntax: <code>show <accounting aps arp as-path bgp chassis class-of-service 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 ospf3 passive monitoring pfe pim policer policy rip ripng route rsvp sap services snmp system task ted version vrrp></code></p>
ssh	<p>Open a secure shell to another host.</p> <p>Syntax: <code>ssh host-name <bypass-routing> <interface interface-name> <inet inet6> <routing instance routing-instance-name> <source source-name> <v1 v2> <vpn-interface vpn-interface-name></code></p>
start	<p>Start a software process.</p> <p>Syntax: <code>start shell</code></p>
telnet	<p>Start a telnet session to another host.</p> <p>Syntax: <code>telnet host-name <8bit> <bypass-routing> < inet inet6> <interface interface-name> <logical-router logical-router-name> <noresolve> <port port-number> <routing-instance routing-instance-name> <source source-address></code></p>
test	<p>Run various diagnostic debugging commands.</p> <p>Syntax: <code>test configuration (filename terminal)</code></p>
traceroute	<p>Trace the route to a remote host.</p> <p>Syntax: <code>traceroute host-name <as-number-lookup> <bypass-routing> <gateway address> <inet><inet6> <logical-router logical-router-name> <noresolve> <routing-instance routing-instance-name> <source address> <tos value> <tll value> <interface interface-name> <wait seconds></code></p>

Using the CLI Operational Mode

This section describes how to use the CLI operational mode. You can do the following:

- Entering the CLI Operational Mode on page 168
- Getting Help on Commands at a Hierarchy Level on page 168
- Getting Help About Commands on page 169
- Having the CLI Complete Commands on page 170
- Using CLI Command Completion on page 170
- Displaying CLI Command History on page 171

Entering the CLI Operational Mode

To enter the JUNOS software CLI, use the following command:

```
user@host> cli
```

You are in the CLI when you see the > prompt, preceded by a string that defaults to the name of the user and the name of the router. For example:

```
user@host>
```

Getting Help on Commands at a Hierarchy Level

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. You have the following options:

- 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.

Getting Help About Commands

To get help about operational mode CLI commands, you can do the following:

- Listing Top-Level Operational Mode CLI Commands on page 169
- Listing CLI Commands That Start with a Particular Letter on page 169
- Listing All Available Commands of a Particular Type on page 169

Listing Top-Level Operational Mode CLI Commands

To list all available commands at the top level of the CLI operational mode, use the following command (see Table 23 on page 166):

```
user@host> ?
```

Possible completions:

clear	Clear information in the system
configure	Manipulate software configuration information
file	Perform file operations
help	Provide help information
monitor	Show real-time debugging information
mtrace	Trace multicast path from source to receiver
ping	Ping remote target
quit	Exit the management session
request	Make system-level requests
restart	Restart software process
set	Set CLI properties, date/time, craft interface message
show	Show system information
ssh	Start secure shell on another host
start	Start shell
telnet	Telnet to another host
test	Perform diagnostic debugging
tracert	Trace route to remote host

Listing CLI Commands That Start with a Particular Letter

To list all commands that start with the letter c, use the following CLI command:

```
user@host> c?
```

Possible completions:

clear	Clear information in the system
configure	Manipulate software configuration information

```
user@host> c
```

Listing All Available Commands of a Particular Type

To list all available clear commands, use the following CLI command:

```
user@host> clear ?
```

Possible completions:

arp	Clear address resolution information
bfd	Clear Bidirectional Forwarding Detection information
bgp	Clear Border Gateway Protocol information
cli	Clear command-line interface settings
esis	Clear end system-to-intermediate system information
firewall	Clear firewall counters

helper	Clear port-forwarding helper information
igmp	Clear Internet Group Management Protocol information
ike	Clear IKE information
ilmi	Clear interim local management interface statistics
interfaces	Clear interface information
ipsec	Clear IP Security information
ipv6	Clear IP version 6 information
isis	Clear Intermediate System-to-Intermediate System information
ldp	Clear Label Distribution Protocol information
log	Clear contents of log file
mld	Clear multicast listener discovery information
mpls	Clear Multiprotocol Label Switching information
msdp	Clear Multicast Source Discovery Protocol information
multicast	Clear multicast information
ospf	Clear Open Shortest Path First information
ospf3	Clear Open Shortest Path First version 3 information
pgm	Clear Pragmatic Generalized Multicast information
pim	Clear Protocol Independent Multicast information
rip	Clear Routing Information Protocol information
ripng	Clear Routing Information Protocol for IPv6 information
rsvp	Clear Resource Reservation Protocol information
	services
snmp	Clear Simple Network Management Protocol information
system	Clear system information
vpls	Clear VPLS information
vrrp	Clear Virtual Router Redundancy Protocol statistics

Having 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 by a question mark.

To complete a command or option that you have partially typed, press the **Tab** key or the spacebar. 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, and the possible completions are displayed.

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

Using CLI Command Completion

To complete the `show interfaces` command, do the following:

```
user@host> show in<Spacebar>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
[...Output truncated...]
```

To display a list of all log files whose names start with the string “messages,” and then display the contents of one of the files, do the following:

```
user@host> 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@host> show log mes<Tab>sages.4<Tab>.gz<Enter>
Jan 15 21:00:00 myhost newsyslog[1381]: logfile turned over
[...Output truncated...]
```

Displaying 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. Specify a number with the command to display 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
```

CLI Configuration Mode

In configuration mode, you configure the JUNOS software by creating a hierarchy of configuration statements. You can do this by 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 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 24 explains each CLI configuration mode command. The commands are organized alphabetically.

Table 24: CLI Configuration Mode Commands

Command	Description
activate	Remove the <code>inactive:</code> 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 <code>commit</code> command. Syntax: <code>activate (statement-path identifier)</code>
annotate	Add comments to a configuration. Syntax: <code>annotate statement-path "comment-string"</code>
commit	Commit the set of changes to the database and cause the changes to take operational effect. Syntax: <code>commit <<at <string>> <and-quit> <check> <confirmed <minutes>> <synchronize></code>
copy	Make a copy of an existing statement in the configuration. Syntax: <code>copy existing-statement-path to new-statement-path</code>
deactivate	Add the <code>inactive:</code> tag to a statement, effectively commenting out the statement or identifier from the configuration. Statements or identifiers marked as <code>inactive</code> do not take effect when you issue the <code>commit</code> command. Syntax: <code>deactivate (statement-path identifier)</code>
delete	Delete a statement or identifier. All subordinate statements and identifiers contained within the specified statement path are deleted with it. Syntax: <code>delete (statement-path identifier)</code>
edit	Move inside the specified statement hierarchy. If the statement does not exist, it is created. Syntax: <code>edit statement-path</code>
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 <code>quit</code> and <code>exit</code> commands are synonyms. Syntax: <code>exit (configuration-mode)</code>
help	Display help about available configuration statements. Syntax: <code>help (apropos topic reference) <string></code>
insert	Insert an identifier into an existing hierarchy. Syntax: <code>insert (statement-path) identifier1 (before after) identifier2</code>

Command	Description
load	<p>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.</p> <p>Syntax: <code>load (merge override patch replace) (filename terminal)</code></p>
quit	<p>Exit the current level of the statement hierarchy, returning to the level prior to the last edit command, or exit from configuration mode. The <code>quit</code> and <code>exit</code> commands are synonyms.</p> <p>Syntax: <code>quit configuration-mode</code></p>
rename	<p>Rename an existing configuration statement or identifier.</p> <p>Syntax: <code>rename <statement-path> identifier1 to identifier2</code></p>
rollback	<p>Return to a previously committed configuration. The software saves the last 50 committed configurations, including the rollback number, date, time, and name of the user who issued the commit configuration command.</p> <p><code>rollback 0</code> erases any configuration changes made to the current candidate configuration. The currently operational JUNOS software configuration is stored in the file <code>juniper.conf</code>, and the last three committed configurations are stored in the files <code>juniper.conf.1.gz</code>, <code>juniper.conf.2.gz</code>, and <code>juniper.conf.3.gz</code>. These four files are located in the directory <code>/config/</code>, which is on the router's flash drive. The remaining 46 previous committed configurations, the files <code>juniper.conf.4.gz</code> through <code>juniper.conf.49.gz</code>, are stored in the directory <code>/var/db/config/</code>, which is on the router's hard disk.</p> <p>Syntax: <code>rollback <number></code></p>
run	<p>Run a CLI command without exiting from configuration mode.</p> <p>Syntax: <code>run command</code></p>
save	<p>Save the configuration to an ASCII file, by default in the users home directory. The contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it. This allows a section of the configuration to be saved, while fully specifying the statement hierarchy.</p> <p>Syntax: <code>save filename</code></p>
set	<p>Create a statement hierarchy and set identifier values. This is similar to <code>edit</code> except that your current level in the hierarchy does not change, and you can set identifier values whereas <code>edit</code> only allows access to a statement-path.</p> <p>Syntax: <code>set (statement-path identifier)</code></p>
show	<p>Display the current configuration.</p> <p>Syntax: <code>show (statement-path identifier)</code></p>
status	<p>Display the users currently editing the configuration.</p>
top	<p>Return to the top level of configuration command mode, which is indicated by the <code>[edit]</code> banner, or execute a command from the top level of the configuration.</p> <p>Syntax: <code>top <configuration-command></code></p>
up	<p>Move up one level in the statement hierarchy.</p> <p>Syntax: <code>up <number></code></p>
update	<p>Update a private database.</p>

Configuration Statements and Identifiers

You configure all router properties by including statements in the configuration. A statement consists of a keyword, which is fixed text, and, optionally, an identifier. An identifier is an identifying name that you define, such as the name of an interface, or a username, which allows you and the CLI to discriminate among a collection of statements.

The following list shows the statements available at the top level of the configuration mode (that is, the trunk of the hierarchy tree). Table 25 on page 175 describes each statement.

```
[edit]
user@host# set ?
Possible completions:
> access           Network access configuration
> accounting-options Accounting data configuration
> applications     Define applications by protocol characteristics
+ apply-groups     Groups from which to inherit configuration data
> chassis          Chassis configuration
> class-of-service Class-of-service configuration
> firewall         Define a firewall configuration
> forwarding-options Configure options to control packet sampling
> groups           Configuration groups
> interfaces       Interface configuration
> isdn             ISDN process configuration
> logical-routers  Logical routers
> policy-options   Routing policy option configuration
> protocols        Routing protocol configuration
> routing-instances Routing instance configuration
> routing-options  Protocol-independent routing option configuration
> security         Security configuration
> services         Service PIC applications settings
> snmp            Simple Network Management Protocol configuration
> system          System parameters
```

An angle bracket (>) before the statement name indicates that it is a container statement and you can define other statements at levels below it.

If there is no angle bracket (>) before the statement name, the statement is a leaf statement; you cannot define other statements at hierarchy levels below it.

A plus sign (+) before the statement name indicates that it can contain a set of values. To specify a set, include the values in brackets. For example:

```
[edit]
user@host# set policy-options community my-as1-transit members [65535:10 65535:11]
```

In some statements, you can include an identifier. For some identifiers, such as interface names, you must specify the identifier in a precise format. For example, the interface name **so-0/0/0** refers to a SONET/SDH interface that is on the Flexible PIC Concentrator (FPC) in slot 0, in the first Physical Interface Card (PIC) location, and in the first port on the PIC. For other identifiers, such as interface descriptive text, policy, and firewall term names, you can specify any name, including special characters, spaces, and tabs.

You must enclose in quotation marks (double quotes) identifiers and any strings that include the following characters: space tab () [] { } ! @ # \$ % ^ & | ' = ?

Table 25 describes each top-level CLI configuration mode statement.

Table 25: Configuration Mode Top-Level Statements

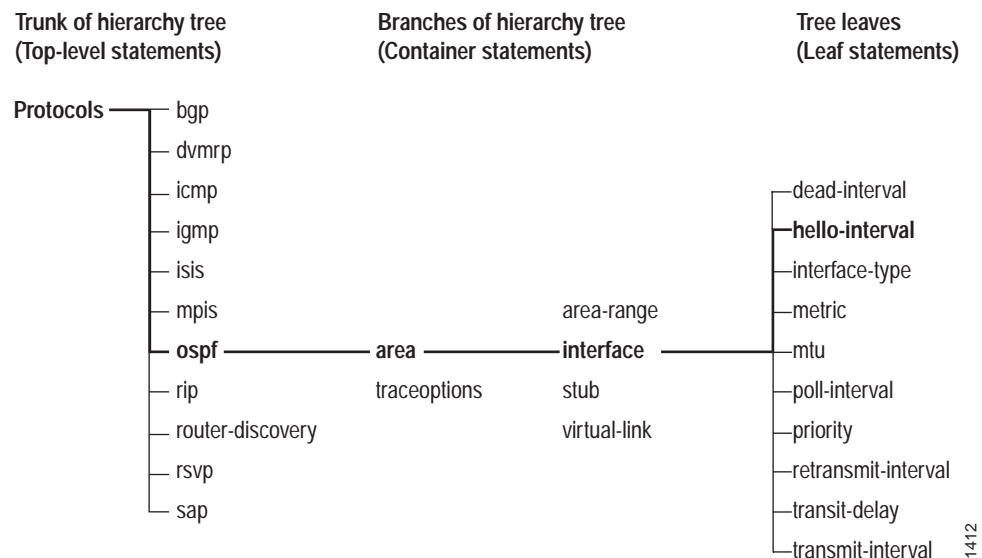
Statement	Description
accounting-options	Configure accounting statistics data collection for interfaces and firewall filters. For information about the statements in this hierarchy, see the <i>JUNOS Network Management Configuration Guide</i> .
chassis	Configure properties of the router chassis, including the clock source, conditions that activate alarms, and SONET/SDH framing and concatenation properties. For information about the statements in this hierarchy, see the <i>JUNOS Network Interfaces Configuration Guide</i> and the <i>JUNOS Class of Service Configuration Guide</i> .
class-of-service	Configure class-of-service parameters. For information about the statements in this hierarchy, see the <i>JUNOS Network Interfaces Configuration Guide</i> and the <i>JUNOS Class of Service Configuration Guide</i> .
firewall	Define filters that select packets based on their contents. For information about the statements in this hierarchy, see the <i>JUNOS Policy Framework Configuration Guide</i> .
forwarding-options	Define forwarding options, including traffic sampling options. For information about the statements in this hierarchy, see the <i>JUNOS Network Interfaces Configuration Guide</i> and the <i>JUNOS Class of Service Configuration Guide</i> .
groups	Configure configuration groups.
interfaces	Configure interface information, such as encapsulation, interfaces, virtual channel identifiers (VCIs), and data-link channel identifiers (DLCIs). For information about the statements in this hierarchy, see the <i>JUNOS Network Interfaces Configuration Guide</i> and the <i>JUNOS Class of Service Configuration Guide</i> .
policy-options	Define routing policies, which allow you to filter and set properties in incoming and outgoing routes. For information about the statements in this hierarchy, see the <i>JUNOS Routing Protocols Configuration Guide</i> .
protocols	Configure routing protocols, including Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), Routing Information Protocol (RIP), Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP), and Resource Reservation Protocol (RSVP). For information about the statements in this hierarchy, see the chapters that discuss how to configure the individual routing protocols in the <i>JUNOS Routing Protocols Configuration Guide</i> and the <i>JUNOS MPLS Applications Configuration Guide</i> .
routing-instances	Configure multiple routing instances. For information about the statements in this hierarchy, see the <i>JUNOS Routing Protocols Configuration Guide</i> .

Statement	Description
routing-options	Configure protocol-independent routing options, such as static routes, autonomous system (AS) numbers, confederation members, and global tracing (debugging) operations to log. For information about the statements in this hierarchy, see the <i>JUNOS Routing Protocols Configuration Guide</i> .
snmp	Configure Simple Network Management Protocol (SNMP) community strings, interfaces, traps, and notifications. For information about the statements in this hierarchy, see the <i>JUNOS Network Management Configuration Guide</i> .
system	Configure systemwide properties, including the hostname, domain name, Domain Name System (DNS) server, user logins and permissions, mappings between hostnames and addresses, and software processes.

Configuration Statement Hierarchy

The JUNOS software configuration consists of a hierarchy of *statements*. There are two types of statements: *container statements*, which are statements that contain other statements, and *leaf statements*, which do not contain other statements (see Figure 19). All of the container and leaf statements together form the *configuration hierarchy*.

Figure 19: Configuration Mode Hierarchy of Statements



Each statement at the top level of the configuration hierarchy resides at the trunk (or root level) of a hierarchy tree. The top-level statements are container statements, containing other statements that form the tree branches. The leaf statements are the leaves of the hierarchy tree. An individual hierarchy of statements, which starts at the trunk of the hierarchy tree, is called a *statement path*. Figure 19 illustrates the hierarchy tree, showing a statement path for the portion of the protocol configuration hierarchy that configures the hello interval on an interface in an OSPF area.

The `protocols` statement is a top-level statement at the trunk of the configuration tree. The `ospf`, `area`, and `interface` statements are all subordinate container statements of a higher statement (they are branches of the hierarchy tree), and the `hello-interval` statement is a leaf on the tree, which, in this case, contains a data value: the length of the hello interval, in seconds.

The CLI represents the statement path shown in Figure 19 as [`protocols ospf area area-number interface interface-name`], and displays the configuration as follows:

```

protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0 {
        hello-interval 5;
      }
      interface so-0/0/1 {
        hello-interval 5;
      }
    }
  }
}

```

The CLI indents each level in the hierarchy to indicate each statement's relative position in the hierarchy and generally sets off each level with braces, using an open brace at the beginning of each hierarchy level and a closing brace at the end. If the statement at a hierarchy level is empty, the braces are not printed. Each leaf statement ends with a semicolon. If the hierarchy does not extend as far as a leaf statement, the last statement in the hierarchy ends with a semicolon.

The CLI uses this indented representation when it displays the current system configuration, and you use this format when creating ASCII files that contain the software configuration. However, the format of ASCII configuration files is not as strict as the CLI output of the configuration. Although the braces and semicolons are required, the indentation and use of new lines, as shown above, are not required in ASCII configuration files.

Using the CLI Configuration Mode

This section describes how to use the CLI configuration mode. You can do the following tasks:

- Entering Configuration Mode on page 178
- Exiting Configuration Mode on page 179
- Moving Among Levels of the Hierarchy on page 179
- Displaying the Current Configuration on page 179
- Modifying the Configuration on page 181
- Removing a Statement on page 181
- Running Operational Mode CLI Commands from Configuration Mode on page 181

- Displaying Configuration Mode Command History on page 182
- Committing a Configuration on page 182
- Saving a Configuration to a File on page 183
- Returning to a Previously Committed Configuration on page 183
- Getting Help About Statements on page 185

Entering Configuration Mode

If many users enter configuration mode at the same time, everyone can make configuration changes and commit all changes. If one user enters configuration mode when another user is also in configuration mode, a message indicates who the user is and what portion of the configuration that user is viewing or editing. To enter configuration mode, use the following CLI command:

```
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
```

- If, when you enter configuration mode, the configuration contains changes that have not been committed, a message appears:

```
user@host> configure
```

```
Entering configuration mode
The configuration has been changed but not committed
```

- If, while in configuration mode, you try to make a change while the configuration is locked by another user, a message indicates that the configuration database is locked, who the user is, and what portion of the configuration the user is viewing or editing:

```
[edit]
user@host# set system host-name ipswitch
```

```
error: configuration database locked by:
  user2 terminal d0 (pid 1828) on since 19:47:58 EDT, idle 00:02:11
    exclusive [edit protocols]
```

- If you enter configuration mode with the **configure exclusive** command, you lock the candidate configuration for as long as you remain in configuration mode, allowing you to make changes without interference from other users. If another user is also in configuration mode and has the configuration locked, a message indicates who the user is and what portion of the configuration the user is viewing or editing:

```
user@host> configure exclusive
```

```
Entering configuration mode
Users currently editing the configuration:
```

```

root terminal p3 (pid 1088) on since 2000-10-30 19:47:58 EDT, idle
00:00:44
exclusive [edit interfaces so-3/0/0 unit 0 family inet]

```

Exiting Configuration Mode

To exit configuration mode, use the `exit configuration-mode` configuration mode command from any level or use the `exit` command from the top level. If you try to exit from configuration mode using the `exit` command and the configuration contains changes that have not been committed, you see a message and prompt:

```

[edit]
user@host# exit

```

```

The configuration has been changed but not committed
Exit with uncommitted changes? [yes,no] (yes) <Enter>
Exiting configuration mode
user@host>

```

To exit with uncommitted changes without having to respond to a prompt, use the `exit configuration-mode` command.

Moving Among Levels of the Hierarchy

The CLI commands in Table 26 help you navigate the levels of the configuration statement hierarchy.

Table 26: CLI Configuration Mode Navigation Commands

Command	Description
<code>edit</code>	To move down through an existing configuration command hierarchy, or to create a hierarchy and move down to that level, use the <code>edit</code> configuration mode command, specifying the hierarchy level at which you want to be.
<code>exit</code>	To move up the hierarchy, use the <code>exit</code> configuration mode command. This command is, in effect, the opposite of the <code>edit</code> command.
<code>up</code>	To move up the hierarchy one level at a time, use the <code>up</code> configuration mode command.
<code>top</code>	To move directly to the top level, use the <code>top</code> configuration mode command.

Displaying the Current Configuration

You can display the following information about the current configuration:

- Displaying the Configuration at the Current Hierarchy Level on page 180
- Displaying the Last Committed Current Configuration on page 180
- Displaying Users Currently Editing the Configuration on page 180

Displaying the Configuration at the Current Hierarchy Level

To display the configuration at the current hierarchy level or at the specified level, use the **show** configuration mode command.

```
user@host> show <statement-path>
```

The configuration statements appear in a fixed order. The CLI indents each level in the hierarchy to indicate each statement's relative position in the hierarchy and generally sets off each level with braces, using an open brace at the beginning of each hierarchy level and a closing brace at the end. If the statement at a hierarchy level is empty, the braces are not printed. Each leaf statement ends with a semicolon. If the hierarchy does not extend as far as a leaf statement, the last statement in the hierarchy ends with a semicolon. Interfaces appear alphabetically by type, and then in numerical order by slot number, PIC number, and port number.

Displaying the Last Committed Current Configuration

You also can use the CLI operational mode **show configuration** command to display the last committed current configuration, which is the configuration currently running on the router:

```
user@host> show configuration
```

Displaying Users Currently Editing the Configuration

To display the users currently editing the configuration, use the **status configuration mode** command:

```
[edit]
user@host# status
```

```
Current configuration users:
  user terminal p0 (pid 518) on since 2002-03-12 18:24:27 PST
    [edit protocols]
```

The system displays who is editing the configuration (**user**), how the user is logged in (**terminal p0**), the date and time the user logged in (**2002-03-12 18:24:27 PST**), and what level of the hierarchy the user is editing (**[edit protocols]**).

Modifying the Configuration

To configure the router or to modify an existing router configuration, you add statements to the configuration. For each statement hierarchy, you create the hierarchy starting with a statement at the top level and continuing with statements that move progressively lower in the hierarchy.

To modify the hierarchy, you use two configuration mode commands:

- **set**—Creates a statement hierarchy and sets identifier values. After you issue a **set** command, you remain at the same level in the hierarchy. The **set** command has the following syntax:

```
set <statement-path> statement <identifier>
```

statement-path is the hierarchy to the configuration statement and the statement itself. If you have already moved to the statement's hierarchy level, you omit this. *statement* is the configuration statement itself. *identifier* is a string that identifies an instance of a statement.

- **edit**—Moves to a particular hierarchy level. If that hierarchy level does not exist, the **edit** command creates it and then moves to it. The **edit** command has the following syntax:

```
edit <statement-path> statement <identifier>
```

Removing a Statement

To delete a statement or identifier, use the **delete** configuration mode command. Deleting a statement or an identifier effectively “unconfigures” the functionality associated with that statement or identifier, returning that functionality to its default condition. When you delete a statement, the statement and all its subordinate statements and identifiers are removed from the configuration.

```
delete <statement-path> <identifier>
```

To delete the entire hierarchy starting at the current hierarchy level, do not specify a statement or an identifier in the **delete** command:

```
[edit]
user@host# delete
```

```
Delete everything under this level? [yes, no] (no) ?
```

```
Possible completions:
```

```
no          Don't delete everything under this level
```

```
yes         Delete everything under this level
```

```
Delete everything under this level? [yes, no] (no)
```

Running Operational Mode CLI Commands from Configuration Mode

To display the output of an operational mode **show** or other command while configuring the software, you can execute a single operational mode command by issuing the **run** configuration mode command and specifying the operational mode command:

```
[edit]
user@host# run operational-mode-command
```

Displaying Configuration Mode Command History

To display a list of the recent commands you issued while in configuration mode, use the `run show cli history` command. By default, this command displays the last 100 commands issued in the CLI.

```
[edit]
user@host# run show cli history
```

```
12:40:08 -- show
12:40:17 -- edit protocols
12:40:27 -- set isis
12:40:29 -- edit isis
12:40:40 -- run show cli history
```

Committing a Configuration

To commit a configuration, you can do the following:

- Saving Configuration Changes and Activating the Configuration on page 182
- Saving Configuration Changes, Activating the Configuration, and Exiting Configuration Mode on page 182

Saving Configuration Changes and Activating the Configuration

To save software configuration changes to the configuration database and activate the configuration on the router, use the `commit` configuration mode command:

```
[edit]
user@host# commit
```

```
commit complete
```

The configuration is checked for syntax errors. If the syntax is correct, the configuration is activated and becomes the current, operational router configuration. If the configuration contains syntax errors, a message indicates the location of the error and the configuration is not activated. You must correct the error before recommitting the configuration.

Saving Configuration Changes, Activating the Configuration, and Exiting Configuration Mode

To save software configuration changes, activate the configuration on the router, and exit configuration mode, use the `commit and-quit` configuration mode command. This command succeeds only if the configuration contains no errors.

```
[edit]
user@host# commit and-quit
```

```
commit complete
exiting configuration mode
user@host>
```

Saving a Configuration to a File

To save the configuration to a text (ASCII) file so that you can edit it with a text editor of your choice, use the **save** configuration mode command. By default, the configuration is saved to that file in your home directory, which is on the flash disk.

```
[edit]
user@host# save filename
```

Wrote 475 lines of configuration to '*filename*'

Returning to a Previously Committed Configuration

To return to a previously committed configuration, you can do the following:

- Returning to the Most Recently Committed Configuration on page 183
- Activating the Configuration You Loaded on page 183
- Returning to a Configuration Prior to the Most Recently Committed One on page 183
- Displaying Previous Configurations on page 184

Returning to the Most Recently Committed Configuration

To return to the most recently committed configuration and load it into configuration mode without activating it, use the **rollback** configuration mode command:

```
[edit]
user@host# rollback
```

load complete

Activating the Configuration You Loaded

To activate the configuration that you loaded, use the **commit** command:

```
[edit]
user@host# rollback
load complete
[edit]
user@host# commit
```

Returning to a Configuration Prior to the Most Recently Committed One

To return to a configuration prior to the most recently committed one, include the number in the **rollback** command. *number* can be a number in the range 0 through 9. The most recently saved configuration is number 0 (which is the default configuration to which the system returns), and the oldest saved configuration is number 9.

```
[edit]
user@host# rollback number
```

load complete

Displaying Previous Configurations

To display previous configurations, including the rollback number, date, time, the name of the user who committed changes, and the method of commit, use the `rollback ?` command.

```
[edit]
user@host# rollback ?
```

Possible completions:

```
<[Enter]> Execute this command
```

```
<number> Numeric argument
```

```
0      2005-02-27 12:52:10 PST by abc via cli
1      2005-02-26 14:47:42 PST by def via cli
2      2005-02-14 21:55:45 PST by ghi via cli
3      2005-02-10 16:11:30 PST by jkl via cli
4      2005-02-10 16:02:35 PST by mno via cli
5      2005-03-16 15:10:41 PST by pqr via cli
6      2005-03-16 14:54:21 PST by stu via cli
7      2005-03-16 14:51:38 PST by vwx via cli
8      2005-03-16 14:43:29 PST by yzz via cli
9      2005-03-16 14:15:37 PST by abc via cli
10     2005-03-16 14:13:57 PST by def via cli
11     2005-03-16 12:57:19 PST by root via other
12     2005-03-16 10:45:23 PST by root via other
13     2005-03-16 10:08:13 PST by root via other
14     2005-03-16 01:20:56 PST by root via other
15     2005-03-16 00:40:37 PST by ghi via cli
16     2005-03-16 00:39:29 PST by jkl via cli
17     2005-03-16 00:32:36 PST by mno via cli
18     2005-03-16 00:31:17 PST by pqr via cli
19     2005-03-15 19:59:00 PST by stu via cli
20     2005-03-15 19:53:39 PST by vwx via cli
21     2005-03-15 18:07:19 PST by yzz via cli
22     2005-03-15 17:59:03 PST by abc via cli
23     2005-03-15 15:05:14 PST by def via cli
24     2005-03-15 15:04:51 PST by ghi via cli
25     2005-03-15 15:03:42 PST by jkl via cli
26     2005-03-15 15:01:52 PST by mno via cli
27     2005-03-15 14:58:34 PST by pqr via cli
28     2005-03-15 13:09:37 PST by root via other
29     2005-03-12 11:01:20 PST by stu via cli
30     2005-03-12 10:57:35 PST by vwx via cli
31     2005-03-11 10:25:07 PST by yzz via cli
32     2005-03-10 23:40:58 PST by abc via cli
33     2005-03-10 23:40:38 PST by def via cli
34     2005-03-10 23:14:27 PST by ghi via cli
35     2005-03-10 23:10:16 PST by jkl via cli
36     2005-03-10 23:01:51 PST by mno via cli
37     2005-03-10 22:49:57 PST by pqr via cli
38     2005-03-10 22:24:07 PST by stu via cli
39     2005-03-10 22:20:14 PST by vwx via cli
40     2005-03-10 22:16:56 PST by yzz via cli
41     2005-03-10 22:16:41 PST by abc via cli
42     2005-03-10 20:44:00 PST by def via cli
43     2005-03-10 20:43:29 PST by ghi via cli
44     2005-03-10 20:39:14 PST by jkl via cli
45     2005-03-10 20:31:30 PST by root via other
```

```

46                2005-03-10 18:57:01 PST by mno via cli
47                2005-03-10 18:56:18 PST by pqr via cli
48                2005-03-10 18:47:49 PST by stu via cli
49                2005-03-10 18:47:34 PST by vw via cli
| Pipe through a command
[edit]

```

Getting Help About Statements

In configuration mode, you can use the **help** command to display help based on a text string contained in a statement name. This command displays help for statements at the current hierarchy level and below.

```
user@host# help string
```

You can also display help based on a text string contained in a statement name using the **help topic** and **help reference** commands. The **help topic** command displays usage guidelines for the statement, whereas the **help reference** command displays summary information about the statement.

```

user@host# help topic ?
access                Network access control
accounting-options    Accounting data collection
applications          Application protocols
bgp                   Border Gateway Protocol
chassis               Platform
class-of-service      Class of service (CoS)
connections           Circuit cross-connect (CCC)
dvmrp                 Distance Vector Multicast Routing Protocol
firewall              Firewalls and filters
forwarding-options    Packet sampling
groups                Configuration groups
igmp                  Internet Group Management Protocol
interfaces            Interfaces
isis                  Intermediate System-to-Intermediate System
l2circuit              Layer 2 virtual circuits
layer2-vpns           Layer 2 VPNs
layer3-vpns           Layer 3 VPNs
ldp                   Label Distribution Protocol
link-management        Link Management Protocol
logical-routers        Logical routers
mld                   Multicast Listener Discovery protocol
mpls                  Multiprotocol Label Switching
msdp                  Multicast Source Discovery Protocol
ospf                  Open Shortest Path First protocol
pgm                   Pragmatic Generalized Multicast
pim                   Protocol-Independent Multicast and data MDT
policy-options        Routing policy
rip                   Routing Information Protocol
ripng                 Routing Information Protocol Next Generation
rmon                  Remote Monitoring
router-advertisement  Neighbor discovery
router-discovery       Internet Control Message Protocol router discovery
routing-instances      Routing instances
routing-options        Protocol-independent routing options
rsvp                   Resource Reservation Protocol
sap                   Session Advertisement Protocol

```

security	Internet Protocol security (IPSec)
services	Service sets for Adaptive Services PIC
snmp	Simple Network Management Protocol
system	System parameters
vpls	Virtual private LAN service
vpns	Virtual private networks

[edit]

user@help# **help topic access ?**

Possible completions:

examples	
l2tp	Overview of Layer 2 Tunneling Protocol configuration
point-to-point	Overview of Point-to-Point Protocol configuration
radius-disconnect-port	Port number for RADIUS disconnect server
radius-server	RADIUS server configuration
traceoptions	Trace options for access processes
tunnel-profile	Join multilink bundles based on endpoint discriminator

[edit]

user@host# **help topic access point-to-point**

Configuring the Point-to-Point Protocol

To configure the Point-to-Point Protocol (PPP), do the following:

- * Configuring the Challenge Handshake Authentication Protocol
- * Configuring the Authentication Order

user@host# **help reference string**

If you do not type an option for a statement that requires one, a message indicates the type of information expected. In this example, you need to type an area number to complete the command:

[edit]

user@host# **set protocols ospf area<Enter>**

syntax error, expecting <identifier>.

In this example, you need to type a value for the hello interval to complete the command:

[edit]

user@host# **set protocols ospf area 45 interface so-0/0/0**
hello-interval<Enter>

syntax error, expecting <data>

If you have omitted a required statement at a particular hierarchy level, when you attempt to move from that hierarchy level or when you issue the **show** command in configuration mode, a message indicates which statement is missing. For example:

```
[edit protocols pim interface so-0/0/0]
user@host# top
Warning: missing mandatory statement: 'mode'
[edit]
user@host# show
protocols {
  pim {
    interface so-0/0/0 {
      priority 4;
      version 2;
      # Warning: missing mandatory statement(s): 'mode'
    }
  }
}
```

