JUNIPer
NETWORKS

Junos® Fundamentals Series

DAY ONE:
EXPLORING THE JUNOS CLI

It's day one and you have a job to do. Start using the Junos CLI today. In this Day One booklet you're just a few hours away from modifying, saving, and loading configuration files on your Junos device.

By Cathy Gadecki and Michael Scruggs

Day One: Exploring the Junos CLI

This book is for first-time users of the Junos opoerating system and Juniper Networks products. It not only lays the foundation for learning Junos, it also facilitates understanding of subsequent booklets that appear in the Day One series.

The Junos CLI provides you with new tools, shortcuts, and safeguards. Learn about these new features and save yourself hours at the keyboard. The practical Day One format offers straightforward explanations, step-by-step instructions, and lots of examples. And, the "Try It Yourself" sections let you practice what you just read.

> "Clear, concise, yet with enough technical detail to allow users new to Juniper Networks gear to complete the desired tasks on day one and allow more advanced users to have a handy cheat sheet for those tasks that they may not perform everyday. These guides are essential tools for all who use Juniper Networks equipment."
>
> Orin Blomberg, CCNP, CCIP, CCSP, JNCIS-M, JNCIS-ER, JNCIS-FWW

Day One: Exploring the Junos CLI shows you how to:

· Navigate the CLI's operational mode and configuration mode.

· Understand the hierarchies that underlie each mode.

· Get onboard help and use keyboard shortcuts to speed up your work.

· Show device status, alarms, and other helpful information in operational mode.

· Modify, save, and load configuration files with minimal risk to operations.

· Use basic configuration mode commands such as show, set, and delete.

· Capitalize on the safety features of the Junos operating system commit model.

· Prepare system changes in advance.

· Use the shortcuts and tips of experienced users and avoid common problems.

Juniper Networks Day One booklets provide just the information you need on day one. They are written by subject matter experts and engineers who specialize in getting networks up and running. Look for other titles covering high-performance networking solutions at **www.juniper.net/dayone.** This book is available in both PDF and printed formats.

JUNIPer
NETWORKS

Junos® Fundamentals Series

Day One: Exploring the Junos CLI

By Cathy Gadecki and Michael Scruggs

JUNIPer
NETWORKS

About the Authors
Cathy Gadecki is a Senior Product Marketing Manager at Juniper Networks responsible for JUNOS Software. Her publications include co-authoring *JUNOS for Dummies*, a practical guide to JUNOS Software.

Michael Scruggs is a Technical Publications Manager at Juniper Networks.

Welcome to Day One

Day One booklets help you to get started quickly in a new topic with just the information that you need on day one.

They are designed to provide straightforward explanations, step-by-step instructions, and many detailed examples that are easy to follow. The *Try It Yourself* sections let you practice commands, configurations, and other steps on your own at your laptop or terminal. Look for margin tips along the way, including best practices, alerts, short cuts, tips, and notes.

Why *Day One* Booklets?

It's a simple premise – you want to use your Juniper equipment as quickly and effectively as possible. You don't have the time to wade through all the documents that are available to you. You may not even know where to start. All you want to know is what to do on day one. The JUNOS Software Fundamentals Series allows you to get right to it, one booklet at a time.

Show Me

With a focus on doing, Day One booklets include lots of examples for your inspection. Try these examples yourself, if you choose. The examples show screen output in a fixed-width font with the commands that you enter in **boldface**.

Before You Begin

To access the JUNOS CLI, you must first have access to the device itself, either through the out-of-band console port or the in-band management port. As each network is different, it is beyond the scope of this booklet to walk you through the process of logging in to deployed equipment. So before attempting to log in, you need to understand how your network is set up or have physical access to the device.

If you need information on accessing your device, see the *Quick Start* guide for your product at www.juniper.net/techpubs/.

What this Booklet Offers You

Day One: Exploring the JUNOS CLI will help you to understand the commands and mechanics of the command-line interface. Subsequent booklets in the JUNOS Software Fundamentals Series will leverage this understanding and help you complete the set up of your devices to run in your network.

When you're done with this booklet, you'll be able to:

✓ Navigate the CLI's operational mode and configuration mode on any device run by JUNOS Software.

✓ Understand the hierarchies that underlie each mode.

✓ Get onboard help and use keyboard shortcuts to speed up your work.

✓ Show device status, alarms, and other helpful information in operational mode.

✓ Modify, save, and load configuration files with minimal risk to operations.

✓ Use basic configuration mode commands such as `show`, `set`, and `delete`.

✓ Capitalize on the safety features of the JUNOS Software commit model.

✓ Prepare system changes in advance.

✓ Use the shortcuts and tips of experienced users and avoid common problems.

What You Need to Know Before Reading

Day One: Exploring the JUNOS CLI lays the foundation for learning JUNOS Software and understanding later Day One booklets that appear in the JUNOS Software Fundamentals Series.

Day One: Exploring the JUNOS CLI is for first-time users of JUNOS or Juniper Networks products, but it is written so that it might also serve as a reference or refresher for more experienced JUNOS administrators.

An understanding of basic networking concepts will be helpful as you work your way through these pages.

And How to Give Us Feedback

Do you have a comment or a great idea that might work as a Day One booklet? We'd like to hear your constructive comments and critiques. Please, if you have time, drop us an email at dayone@juniper.net.

Jump In!

And for those of you with no patience... Go ahead and jump ahead to your own path of discovery on your new JUNOS device (in the factory-default configuration) by using these four steps.

Step 1: Connect

Connect your management PC to the console port of your new Juniper device using a null-modem or rollover cable, turn on the PC and start its terminal emulation program, and then power up the Juniper box.

Step 2: Log In

At the login prompt, enter root and at the password prompt press Enter. What it looks like with user input in bold:

```
host (ttyd0)
login: root
password:
```

Step 3: Enter the CLI

After you are authenticated, you enter the UNIX shell. To get to the CLI from the prompt, enter cli. What it looks like with user input in bold:

```
root@Amnesiac% cli
```

Step 4: Explore

You're now in the operational mode of the command-line interface. Enter a question mark anywhere in the command hierarchy, and you'll immediately see a list of possible entries. Go ahead and try it, enter ?.

See that? It's the list of the valid commands at the top of the operational mode hierarchy. You can also use the question mark to find out the valid possibilities to complete a command. For example:

```
user@Amnesiac > show i?
```

Displays all the possible show commands that start with the letter i. Look around ... see what else you can discover. What happens when you enter configure?

Chapter 1

Introducing the CLI

The command-line interface (CLI) is the software interface used to access your device. From here you configure the platform, monitor its operations, and adjust the configuration as needed.

If you've operated other networking devices, the JUNOS CLI should seem familiar, but you will also quickly notice that there are some new and different commands. No need to fret. These changes provide a rich set of new tools and safeguards that help you efficiently manage your network and maintain high uptime.

The command-line interface includes lots of shortcuts and commands to get help. Master these shortcuts and commands, and you'll spend much less time pounding away on your keyboard. With just a little effort, you'll soon learn why so many people say that JUNOS saves them time (often lots of it), reduces repetitive tasks, and helps them to avoid mistakes.

J-Web

If you'd prefer to use a web GUI rather than the CLI, take a look at J-Web, the powerful web-based management interface available on JUNOS devices. J-Web lets you perform the same actions available in the command-line interface. It provides practical tools to monitor, configure, troubleshoot, and manage your device.

MORE? Go to the J-Web technical documentation for your device at www. juniper.net/techpubs/.

Outlining the Command Modes

The first step to exploring the JUNOS CLI is to understand its two command modes:

- **Operational mode:** manage and monitor device operations. For example, monitor the status of the device interfaces, check chassis alarms, and upgrade and downgrade the device's operating system.

- **Configuration mode:** configure the device and its interfaces. These include user access, interfaces, protocols, security services, and system hardware properties.

Top Level Mode

2nd Level Modes

3rd Level Modes
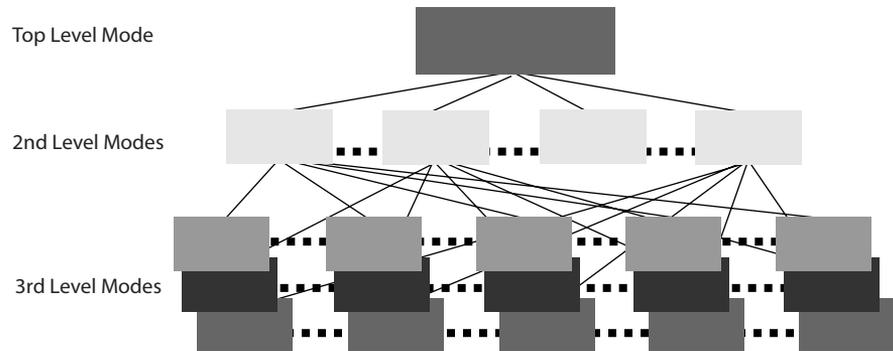
Figure 1.1   Hierarchical Structure of the JUNOS CLI Modes

The JUNOS CLI structures the activities of each mode into hierarchies, as illustrated in Figure 1.1. The hierarchy of each mode is made up of cascading branches of related functions commonly used together.

The structured hierarchy of the command-line interface is one of the many distinctive aspects of the JUNOS CLI preferred by users. By logically grouping activities, the JUNOS CLI provides a regular, consistent syntax helpful for knowing where you are, finding what you want, moving around the interface, and entering commands.

Understanding Operational Command Hierarchies

When you first log in to the CLI, the command-line interface is at the top level of the CLI's operational mode.

On the next page, Figure 1.2 provides a view of the CLI's tree structure from the top of the operational mode, with an example of its cascading hierarchy through the show command. For example, the show configuration hierarchy includes access, chassis, firewall, groups, etc. The structured grouping of commands makes it easy to move quickly up and down the hierarchical path or to a specific function anywhere in the CLI.
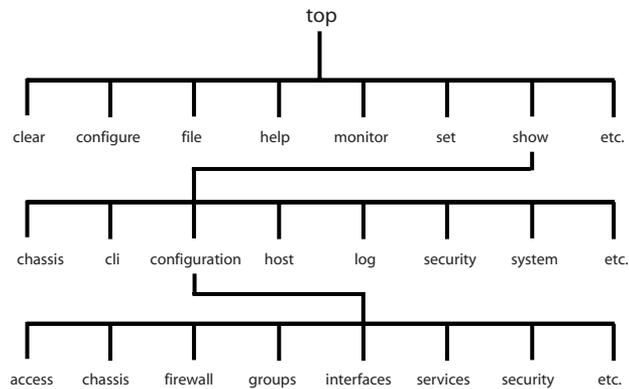
Figure 1.2   **Top of the Operational Mode Tree**

> **NOTE**   The top level of each hierarchy is much like the top of the UNIX filesystem (\), and both the operational mode and configuration mode hierarchies are similar to the directory structure on UNIX systems, PCs, and Macs. You'll learn more about the operational mode in Chapter 3.

Understanding Configuration Statement Hierarchies

Configuration mode has a hierarchical structure logically grouping related configuration statements. This structure eases configuration setup, review, and modification by allowing you to more readily find and view related statements. Figure 1.3 illustrates a portion of the configuration tree, with nodes such as system and interfaces at the second level of the hierarchy.
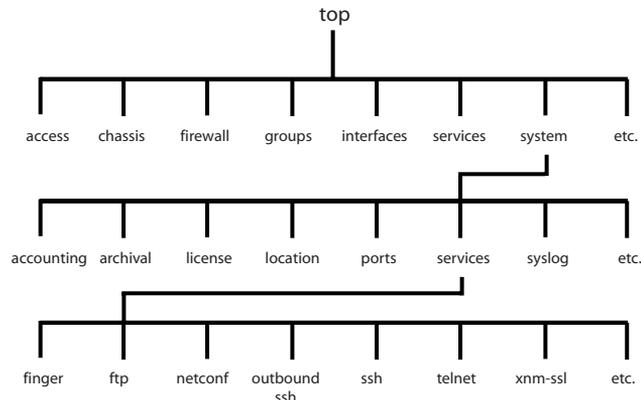
Figure 1.3   **Top of the Configuration Mode Tree**

The configuration statement hierarchy includes two types of statements:

- container statements: contain other statements; that is, they have subordinate configuration levels

- leaf statements: do not contain other statements; they are at the end of a particular hierarchical path

Configuration Syntax

The command-line interface displays the hierarchy of the configuration mode through specific syntax. The following example highlights what you need to know to read a JUNOS CLI configuration listing:

```
[edit]
system {
    services {
        ftp;
    }
}
```

1. The [edit] banner indicates the starting hierarchy level of the listing.

2. The CLI shows the hierarchy of the configuration by indenting each subordinate level.

3. The CLI indicates container statements with open and closed curly braces ({ }). In the above example, system and services are cascading container statements.

4. The CLI indicates leaf statements with a semicolon (;). In the above example, ftp; is a leaf statement.

NOTE   Although the organizational structure within the configuration is similar to C or other programming languages, you do not need to understand programming to understand the configuration file. It simply is an outline view (remember English class) of the configuration. Once you understand how the outline view works, you will find that the configuration is very easy to read and navigate.

Configuration Command Banner

You can always determine where you are in the configuration hierarchy by referring to the configuration command banner, shown as the `[edit]` banner in the example above. When you are in deeper levels of the hierarchy, the `[edit]` banner displays the entire hierarchical path. For example, the banner `[edit system services]` indicates a place of the hierarchy lying within services at the third level, within system at the second level, and within the root first level.

Thus, the following two configuration statements for the FTP service are equal. In the first, you are looking at the statement from the root level of the hierarchy; and so the `ftp;` statement is shown in this listing within the `system` and `services` container statements.

```
[edit]
system {
    services {
        ftp;
    }
}
```

In the second example, you are viewing the `ftp;` statement from deeper within the hierarchy, specifically within the `system` and `services` hierarchies. Because you are deeper within the hierarchy, the command-line only displays the `ftp;` statement.

```
[edit system services]
ftp;
```

The flexibility to work at a specific sublevel in the hierarchy is helpful when you want to focus on just a small portion of the configuration.

You'll learn to navigate through the configuration hierarchy in Chapter 4.

But for now, let's get started using the CLI. It's fast, it's easy, and you can't get lost, because you're using JUNOS Software.

Chapter 2

Getting Started

If you have access to a device in a lab or other nonoperational environment, follow along with the examples on these pages while exploring the CLI. You can enter the commands and examples on your device as well as practice your new skills by using our *Try It Yourself* segments.

To access the CLI, you must connect to the device and then log in. If you need help connecting to your device and logging in, see the Quick Start for your product or go to the URL mentioned below. Before logging in to explore the CLI, you need to understand how your network is set up or have physical access to the device.

The instructions in this Day One booklet also assume that the device's management console has already been configured, and you can log in to the device using a pre-designated user name and password through the management console. This is the standard and recommended method for accessing the CLI on your device.

MORE?   For information on accessing the device out of the box, see the *Quick Start* guide for your product at www.juniper.net/techpubs/.

Logging In

To access the management port from a networked device:

1. Open a command window.

2. If necessary, log in to the gateway server with direct access to the JUNOS device.

    ```
    telnet gatewayserver
    user: username
    password: password
    ```

    Oftentimes, the routers, switches, and security devices are on a subnet behind a gateway router that prevents unauthorized access to these devices.

3. Log in to the device.

    ```
    telnet routername
    user: username
    password: password
    ```

    If the device's IP address is managed by a DNS server, you can simply log in using the designated domain name. Otherwise, you can log in using the unique IP address of the management port.

Switching Between Operational and Configuration Modes

As you monitor and configure a device, you need to switch between the operational mode and the configuration mode. When you change to configuration mode, the command prompt also changes. The operational mode prompt is a right angle bracket (>). The configuration mode prompt is a pound sign (#).

To switch from operational mode to configuration mode, issue the `configure` command:

```
mike@juniper1> configure
```

SHORT CUT When issuing the `configure` command, simply type co. Since no other command starts with those two letters, the CLI will recognize the command and autofill the rest of the command for you.

To exit back to operational mode, issue the `exit configuration-mode` command, or even shorter, the `exit` command.

```
mike@junper1# exit
```

NOTE Keep in mind that if you made configuration changes, you must commit these changes before exiting configuration mode for them to take effect, which is covered in Chapter 4.

Using Keystroke Shortcuts

The JUNOS CLI offers numerous ways to save keystrokes when using the command line, including keyboard sequences and command completion.

All standard UNIX keyboard shortcuts are available to you when you are logged in to the JUNOS device. This is true whether you are in one of the shells, or in the CLI. These shortcuts offer options to shorten keystrokes. It may take a few days for shortened keystrokes to become second nature; however, once you have the muscle memory, these shortcuts can save you lots of typing time.

The CLI stores every entered command in its command history. At any command prompt, the up and down arrow keys let you scroll through this history (on a VT100 terminal type). You can reuse commands that you previously entered, or modify them as needed. Keyboard sequences can save you much time, for example, when you are configuring similar items on the device, or you are repeating operational commands, such as when you are debugging an issue.

<div style="margin-left:2em">

SHORT CUT Use these time-saving keyboard shortcuts:

Go to next in command history	`Down arrow or Ctrl+n`
Go to previous in command history	`Up arrow or Ctrl+p`
Go to beginning of line	`Ctrl+a`
Go to end of line	`Ctrl+e`
Go left one character	`Ctrl+b`
Go right one character	`Ctrl+f`
Go forward one word	`Esc+f`
Go backward one word	`Esc+b`
Delete character over cursor	`Ctrl+d`
Delete word after cursor	`Esc+d`
Delete word before cursor	`Esc+backspace`
Delete text from the cursor to end of the line	`Ctrl+k`
Delete the line	`Ctrl+u`
Paste the deleted text at cursor	`Ctrl+y`

</div>

Command Completion

The CLI provides command completion to further speed your typing in both modes. Command completion automatically finishes partially typed commands, filenames, and user names, so you don't need to recall the exact syntax of the desired input string. Command completion is a big help to new users, easing their transition to the new command-line interface.

The spacebar completes most CLI commands. The tab key not only completes CLI commands, but also filenames and user-defined variables such as policy names, community names, and IP addresses. When the completion of the command or argument is ambiguous, pressing the spacebar or tab key lists the possible completions:

```
[edit]
mike@juniper1# show i<space>
'i' is ambiguous

Possible completions:
```

```
igmp            Show Internet Group Management Protocol
ike             Show Interface Key Exchange Information
interfaces      Show Interface Information
ipsec           Show IP Security Information
isis            Show Intermediate System-to Intermediate
```

SHORT CUT Common abbreviations from other operating systems, such as sh int, are available in JUNOS Software. For example mike@juniper1> **sh<space>ow int<enter>**.

Try It Yourself: Using the Spacebar and Tab Key

Enter the following commands, using the spacebar to complete them:

```
sh<space>ow ro<space>ute
sh<space>ow ch<space>assis h<space>ardware
sh<space>ow conf<space>iguration
cl<space>ear rip s<space>tastics
res<space>tart ro<space>uting g<space>racefully
```

Getting Help

The JUNOS CLI includes several options for getting help any time you're not sure what to do, or if you just want to double-check your memory. Everyone uses the CLI's comprehensive system of online help, even the experts who've been working with Juniper devices for years.

Context-Sensitive Help

Query the command line with the question mark (**?**) character at any level of the operational or configuration hierarchies for a list of available commands and a short description of each. Typing a partial command and ? provides a list of all the valid ways to complete your command. Using ? in either of these ways is known as context-sensitive help in JUNOS lingo:

```
[edit system]
mike@juniper1# set s?
Possible completions:
        saved-core-context      Save context information for core files
        saved-core-files        Number of saved core files per executable (1..64)
      > services               System services
```

```
> static-host-mapping    Static hostname database mapping
> syslog                 System logging facility
```

Try It Yourself: Getting help with a question mark

Display possible completions for the following commands in operational mode:

```
show ?
show chassis ?
show interfaces ?
show system ?
request ?
request support ?
restart ?
ping ?
traceroute ?
```

Display possible completions for the following partially entered commands:

```
s ?
show i ?
request system s ?
restart s ?
```

For commands that require a filename as an argument, the question mark lists the files in the working directory:

```
mike@juniper1> request system license add ?
Possible completions:
 <filename>  Filename (URL, local, remote, or floppy)
 file1      Size: 19701, Last changed: Feb 23 21:56:52
 file2      Size: 1835, Last changed: Apr 09 09:51:57
 log1       Size: 1215, Last changed: Feb 16 13:07:49
 log2       Size: 1135, Last changed: Apr 09 11:05:16
 terminal   Use login terminal
```

Specifying a path lists the files in that directory:

```
mike@junper1> request system license add /cf/ ?
Possible completions:
 <[Enter]>      Execute this command
 <filename>     Filename (URL, local, remote, or floppy)
 /cf/boot/      Last changed: Apr 16 11:08:56
 /cf/dev/       Last changed: Apr 08 2004
 /cf/etc/       Last changed: Apr 30 08:40:09
 /cf/kernel     Size: 32797835, Last changed: Apr 15
 /cf/kernel.old  Size: 32715591, Last changed: Nov 09
 /cf/opt/       Last changed: Nov 09 02:08:43
 /cf/packages/   Last changed: Apr 16 11:08:57
```

```
                            /cf/root/     Last changed: Apr 16 11:08:56
                            /cf/sbin/     Last changed: Apr 16 11:08:56
                            /cf/usr/      Last changed: Nov 09 02:11:23
                            /cf/var/      Last changed: Nov 09 02:11:23
```

Onboard Documentation

When you want more information than what is provided by context-sensitive help, turn to the JUNOS Software technical documentation on your device through the `help` commands. Juniper loads documentation on new devices and includes it as a part of new upgrade builds.

The help files are divided into five major categories. You can access these files in both operational and configuration modes:

- **help apropos**: displays help about a text string contained in a statement or command name

- **help reference**: provides assistance with configuration syntax by displaying summary information for the statement

- **help syslog**: displays information on specific syslog events

- **help tip**: provides random tips for using the CLI

- **help topic**: displays usage guidelines for configuration statements

When requesting help, follow each of the above commands with the string or topic for which you're seeking information.

The Help Apropos Command

The `help apropos` command is useful whenever you remember a portion of a command but not the full statement. The command looks for all matches in statement or command names as well as help strings that are displayed for these:

```
[edit]
mike@junper1# help apropos host-name
set system host-name <host-name>
    Hostname for this router
set system static-host-mapping <host-name>
    Fully qualified name of system
set system services dhcp static-binding <mac-address> host-name <host-name>
    Hostname for this client
set system syslog host
    Host to be notified
set interfaces <interface_name> services-options syslog host <host-name>
    Name of host to notify
```

```
set accounting-options routing-engine-profile <profile-name> fields host-name
    Hostname for this router
set services l2tp tunnel-group <name> syslog host <host-name>
    Name of host to notify
set services service-set <service-set-name> syslog host <host-name>
    Name of host to notify
```

If the string contains spaces, enclose them in quotation marks (" ").

The Help Topic Command

Use the `help topic` command to learn about the usage guidelines for a specific configuration statement:

```
mike@juniper1> help topic interfaces address ?
Configuring the Interface Address

You assign an address to an interface by specifying the address when configuring the
protocol family. For the inet family, you configure the interface's IP address. For the
iso family, you configure one or more addresses for the loopback interface. For the
ccc, tcc, mpls, tnp, and vpls families, you never configure an address.
```

The Help Reference Command

After learning about what a certain command does and when to use it, you can view the actual syntax and possible options using the `help reference` command. Using the same example:

```
mike@juniper1> help reference interfaces address
address
  Syntax

 address address {
    arp ip-address (mac | multicast-mac) mac-address <publish>;
    broadcast address;
    destination address;
    destination-profile name;
    eui-64;
    multipoint-destination address dlci dlci-identifier;
    ...
 Hierarchy Level


  [edit interfaces interface-name unit logical-unit-number family family],
  [edit logical-routers logical-router-name interfaces interface-name unit
  logical-unit-number family family]

  Description

 Configure the interface address.
   <...>
```

NOTE   The `help reference` command is similar to UNIX manpages as well as to the `manual` command seen on other operating systems.

Syntax Help

Rather than waiting until you hit return at the end of a configuration statement, JUNOS Software checks syntax word-by-word. Every time you enter a word into a line and press the spacebar, the CLI determines if each term is a valid command component and whether it is being used properly. If it finds a mistake, the CLI requests correction.

Additionally, JUNOS Software checks for omitted statements required at a particular hierarchy level whenever you attempt to move from that hierarchy level or when you issue the `show` command in configuration mode:

```
 [edit]
mike@juniper1# show
protocols {
        pim {
                interface so-0/0/0 {
                        priority 4;
                        version 2;
                        # Warning: missing mandatory
statement(s): 'mode'
                }
        }
}
```

Filtering Output with the Pipe Command and the More Prompt

You can change how the CLI displays output with the pipe | character and the `more` prompt.

The Pipe Character

The pipe | character lets you filter output in both operational and configuration modes. Pipe makes it possible to display specific information in a single command step, sending the output of one command as input to another, or redirecting the output to a file. The output of the command to the left of the pipe symbol serves as input to the command or file to the right of the pipe.

You can query the CLI to find valid ways to pipe a command, as in this operational mode listing:

```
mike@juniper1> show route | ?
```

```
Possible completions:
  count        Count occurrences
  display      Show additional kinds of information
  except       Show only text that does not match a pattern
  find         Search for first occurrence of pattern
  hold         Hold text without exiting the --More-- prompt
  last         Display end of output only
  match        Show only text that matches a pattern
  no-more      Don't paginate output
  request      Make system-level requests
  resolve      Resolve IP addresses
  save         Save output text to file
  trim         Trim specified number of columns from start of line
```

Using pipes

The following examples from a configured device further demonstrate ways that pipe can help you to fine-tune commands.

Filter command output to a file

Create a file that stores the output of the `request support information` command of the operational mode by piping its output to a `<filename>`:

```
mike@juniper1> request support information | save <filename>
Wrote 1143 lines of output to 'filename'
```

NOTE   See the section *Using the File Commands* in Chapter 3 to learn about accessing the created file.

Display additional and hold information

You can request that a listing include additional information or that the CLI hold information:

■ **| count:** gives the number of lines in the output:

```
mike@juniper1> show interfaces terse | count
Count: 22 lines
```

■ **| display detail:** provides additional information about the contents of the configuration (available only in configuration mode)

■ **| display xml:** shows the output in XML format

```
mike@juniper1> show cli directory | display xml
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/9.410/
junos>
    <cli>
        <working-directory>/var/home/username</working-
directory>
    </cli>
    <cli>
        <banner<master:0></banner>
    </cli>
</rpc-reply>
```

NOTE   It's useful to display output in XML when exchanging configuration and state information with other systems. The XML output is formatted in the standard Remote Procedure Call (RPC) format.

■ **| hold:** retains the output in the buffer until cleared

Constrain command output

The most common way to use the pipe | symbol is to constrain the output:

■ **| match:** specify exactly what information you want to display:

```
mike@juniper1 > show configuration | match at-
at-2/1/0 {
at-2/1/1 {
at-2/2/0 {
at-5/2/0 {
   at-5/3/0 {
```

NOTE   Match is equivalent to the UNIX grep command.

■ **| except:** displays output that ignores a specific string:

```
mike@juniper1> show system users | except root
8:28PM up 1 day, 13:59, 2 users, load averages:
     0.01, 0.01, 0.00
USER  TTY FROM     LOGIN@ IDLE WHAT
sheep p0 baa.juniper.net 7:25PM  - cli
```

■ **| find:** displays the output starting at the first occurrence of the matching text:

```
mike@juniper1>show ethernet-switching interfaces detail | find "Index: 80"
Interface: ge-0/0/16.0 Index: 80
  State: down
  VLANs:
    default             untagged     blocked - blocked by STP

Interface: ge-0/0/17.0 Index: 81
  State: down
  VLANs:
    default             untagged     blocked - blocked by STP
```

■ **| last**: provides only the last screen of the listing

NOTE   When using find or match, you must enclose spaces, operators, or wildcard characters that are a part of the search term in quotation marks.

Multiple Pipes

JUNOS Software sees multiple pipes as a logical AND, only displaying the output that matches all entered pipes. You can enter different pipe commands, as well as the same pipe command, multiple times. For example, to count how many fast Ethernet interfaces are configured within the active configuration:

```
mike@juniper1> show interfaces terse | match fe- | count
Count: 12 lines
```

As another example, use the same pipe command on a single line to show all routes that include the 10.0 string with a /32 subnet mask:

```
mike@juniper1> show route | match /32 | match 10.0
10.0.15.2/32       *[Local/0] 03:18:28
10.0.16.1/32       *[Local/0] 03:20:49
10.0.0.4/32        *[Local/0] 08:54:55
192.168.10.0/32     *[Local/0] 08:57:26
```

The <more> Prompt

The command-line interface automatically paginates output. The CLI settings determine the length for your user account, with the typical setting at 24 lines. When the device stops at a page break, the command-line interface displays the <more> prompt and shows the amount

of displayed output as a percentage of all the content available for display. You can press the h key at any <more> prompt to see a list of display options, such as moving forward and backward in the output, searching, and saving:

```
mike@juniper1> show ethernet-switching interfaces detail
Interface: ge-0/0/0.0 Index: 64
  State: down
  VLANs:
    default             untagged     blocked - blocked by STP

*// Data Deleted From Example //*

Interface: ge-0/0/12.0 Index: 76
  State: down
  VLANs:
    default             untagged     blocked - blocked by STP

---<more>--- h

---(Help for CLI automore)---
        Clear all match and except strings:      c or C
    Display all line matching a regexp             m or M <string>
    Display all lines except those matching a regexp: e or E <string>
    Display this help text:                      h
    Don't hold in automore at bottom of output:  N
    Hold in automore at bottom of output:        H
    Move down half display:                      TAB, d, or ^D
    Move down one line:                          Enter, j, ^N, ^X, ^Z, or Down-Arrow
    Move down one page:                          Space, f, ^F, or Right-Arrow
    Move to bottom of output:                     G, ^E, or End
    Move to top of output:                       g, ^A, or Home
    Move up half display:                        u or ^U
    Move up one line:                            k, Delete, Backspace, ^P, or Up-Arrow
    Move up one page:                            b, ^B, or Left-Arrow
    Quit automore:                               q, Q, ^K
    Redraw display:                              ^L or ^R
    Repeat a keystroke command 1 to 9 times:     Meta-1..9
    Repeat last search:                          n
    Save output to a file:                       s or S <filename/url>
    Search backwards thru the output:            ?<string>
    Search forwards thru the output:              /<string>
---(End of Help)---
```

TIP   The set cli screen-length command modifies the number of displayed lines. Alternatively, you can display the entire output by adding the pipe | no-more as part of your command.

Working with Shells

A *shell* normally refers to any command-line interface used on a UNIX or UNIX-like operating system. These interfaces interpret high-level commands into low-level instructions that the device can use. When you log in to a JUNOS device, you are placed into one of two shells: the C shell or the CLI in operational mode.

The following shells are available:

Shell	Command Prompt
C shell	**%**
Bourne shell	**$**
CLI	**mike@juniper1>** (operational mode) **mike@juniper1#** (configuration mode)

You can navigate from one shell to another by issuing the proper command. This command spawns a new instance of that shell and places you into that instance. For example, while in the CLI operational mode, you can spawn a new C shell interface by issuing the `start shell` command.

ALERT! When you log out of the device be sure to exit each of the shells you entered during your session.

Logging In to the CLI

To log in to the CLI interface, issue the `cli` command at any shell prompt:

```
% cli
```

The CLI always opens in operational mode.

Logging In to the C Shell

To log in to the C shell interface from the CLI in operational mode, issue the `start shell` command:

```
mike@juniper1> start shell
```

To log in to the C shell interface from the CLI in configuration mode, issue the run start shell command:

```
mike@juniper1# run start shell
```

SHORTCUT   The run command allows you to issue CLI operational mode commands while in configuration mode. Just add the keyword run before any operational mode command that you want to execute while you are inside the configuration mode.

To log in to the C shell interface from the Bourne shell interface, issue the csh command.

```
$ csh
```

Logging In to the Bourne Shell

To log in to the Bourne shell interface from the CLI in operational mode, issue the start shell command:

```
mike@juniper1> start shell
```

To log in to the C shell interface from the CLI in configuration mode, issue the run start shell sh command:

```
mike@juniper1# run start shell sh
```

To log in to the C shell interface from the C shell interface, issue the sh command.

```
% sh
```

Logging Out

When logging out, you must log out of each shell you have opened before you can log out completely from the device. Thus if you log in to the device and are placed in the CLI operational mode and then you enter the CLI configuration mode, you need to exit out of both shells.

When it's time to take a break, you must log out of each shell you have opened before you log out comletely from the device. When you are completely logged out of the device, you will receive the message: "Connection closed by foreign host."

```
mike@junper1> exit
logout
```

```
Connection closed by foreign host.
$
```

If you're in configuration mode and want to log out, exit your configuration session to enter operational mode:

```
[edit protocols ospf]
mike@juniper1# exit configuration-mode
Exiting configuration mode

mike@juniper1> exit
logout
Connection closed by foreign host.
$
```

BEST PRACTICE   Protect the security of your device by logging out if you have no reason to be logged in or when you are away from your terminal, even for a few minutes. This prevents someone else from sitting down at your workstation and inadvertently (or deliberately) accessing the device.

Chapter 3

Understanding Operational Mode

Operational mode provides commands for monitoring, managing, and maintaining your device. You can find out the status of your device, administer diagnostics, and perform other operational tasks, as well as manage the software running the device.

Key operational mode capabilities include:

- Monitoring and troubleshooting the device

- Connecting to other network systems

- Restarting software processes

- Entering configuration mode

- Displaying the configuration

- Controlling the CLI environment

- Performing system-level operations such as stopping and rebooting the device and loading JUNOS Software images

JUNOS Software provides an extensive set of on-board instrumentation capabilities for gathering critical operational status, statistics, and other information. These tools deliver advance notification of issues and speed problem solving during events.

As part of your configuration setup you can specify the types of events to track, the event severity, and the files in which to store the data, among other options. Juniper devices come with sufficient processing power to collect and store critical operational data, including SNMP management, system logging, and traceoptions that help you to understand how the box operates in normal conditions and where, when, and why changes occur.

MORE? Find out more about configuring basic monitoring functions for your device in the upcoming booklets of the *JUNOS Software Fundamentals Series*. Download new titles as they become available at www.juniper.net/dayone.

Looking at Operational Mode

Explore operational mode from the top level of its hierarchy. Here's a truncated listing of its most commonly used commands:

```
mike@juniper1> ?
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
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
traceroute      Trace route to remote host
<snip>
```

Showing Device Status

Operational mode provides a large group of show commands to display status and statistics for just about everything on the device:

```
mike@juniper1> show?
Possible completions:
accounting       Show accounting profiles and records
aps              Show Automatic Protection Switching information
arp              Show system Address Resolution Protocol table entries
as-path          Show table of known autonomous system paths
bfd Show         Bidirectional Forwarding Detection information
bgp Show         Border Gateway Protocol information
chassis          Show chassis information
class-of-service Show class-of-service (CoS) information
cli              Show command-line interface settings
configuration    Show current configuration
connections      Show circuit cross-connect connections
dialer           Show dialer information
dlsw             Show DLSw information
dvmrp            Show Distance Vector Multicast Routing Protocolinformation
dynamic-tunnels  Show dynamic tunnel information information
esis             Show end system-to-intermediate system information
firewall         Show firewall information
helper           Show port-forwarding helper information
host             Show hostname information from domain name server
igmp             Show Internet Group Management Protocol information
ike              Show Internet Key Exchange information
interfaces       Show interface information
ipsec            Show IP Security information
```

```
ipv6              Show IP version 6 information
isdn              Show Integrated Services Digital Network information
isis              Show Intermediate System-to-Intermediate System info...
l2circuit         Show Layer 2 circuit information
l2vpn             Show Layer 2 VPN information
ldp               Show Label Distribution Protocol information
llc2              Show LLC2 protocol related information
log               Show contents of log file
mld               Show multicast listener discovery information
mpls              Show Multiprotocol Label Switching information
msdp              Show Multicast Source Discovery Protocol information
multicast         Show multicast information
ntp               Show Network Time Protocol information
ospf              Show Open Shortest Path First information
ospf3             Show Open Shortest Path First version 3 information
pfe               Show Packet Forwarding Engine information
pgm               Show Pragmatic Generalized Multicast information
pim               Show Protocol Independent Multicast information
policer           Show interface policer counters and information
policy            Show policy information
pppoe             Show PPP over Ethernet information
rip               Show Routing Information Protocol information
ripng             Show Routing Information Protocol for IPv6 information
route             Show routing table information
rsvp              Show Resource Reservation Protocol information
sap               Show Session Announcement Protocol information security
services          Show services information
snmp              Show Simple Network Management Protocol information
system            Show system information
task              Show routing protocol per-task information
ted               Show Traffic Engineering Database information
version           Show software process revision levels
vrrp              Show Virtual Router Redundancy Protocol information
```

TIP For the reader with experience using Cisco IOS software, a basic difference of JUNOS Software is that it does not use the keyword IP. So, many of the show commands you already know work if you drop this part of the command. For example, the IOS command show ip route simply becomes show route in JUNOS.

The show command includes other arguments to modify the output. For example, below are the available arguments for the show interfaces command for the fe-1/1/1 Fast Ethernet interface:

```
mike@juniper1> show interfaces fe-1/1/1 ?
Possible completions:
<[Enter]>         Execute this command
brief             Display brief output
```

```
descriptions      Display interface description strings
detail            Display detailed output
extensive         Display extensive output
media             Display media information
snmp-index        SNMP index of interface
statistics        Display statistics and detailed output
terse             Display terse output
<snip>
```

You can add these options to adjust the output listings to what you need. Compare the show output when adding brief and terse to the command above.

```
mike@juniper1> show interfaces fe-1/1/1 brief
Physical interface: fe-1/1/1 Enabled, Physic link is Down
 Link-level type:Ethernet, MTU: 1514, Spped: 100mbps, Loopback:
          Disabled, Source filtering: Disabled
 Flow control  : Enabled
 Device flags  : Present Running Down
 Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
 Link flags    : None

 mike@juniper1> show interfaces fe-1/1/1 terse
 Interface    Admin Link Proto Local Remote
 fe-1/1/1   up  up
 at-1/3/0.0   up  up inet 1.0.0.1 --> 1.0.0.2
          iso
```

TIP   The clear commands let you reset the device's statistics to zero.

Managing Basic Operations

JUNOS Software supports standard network utilities and remote access for management. You may recognize a few of these fundamental commands from UNIX and other operating systems:

- ping: this standard IP command tests whether other devices, interface cards, or nodes are reachable on the network.

- traceroute: this network utility reports the path taken by packets from your device to a destination on an IP network.

- ssh: this standard UNIX secure shell program opens a user shell on another device or host on the network.

- telnet: this management protocol opens a terminal connection to another device or host on the network.

Using the File Commands

The `file` commands let you view and copy files from one location of your device to another, from your device to a remote system, such as a server, or from a remote system to the device. Saving and loading configuration files on the device are helpful for:

- Archiving and backing up configurations

- Sharing configuration files across devices

- Saving and loading parts of configuration files that might be common across many devices within a network (route filters, for instance). To view a file use the `file show` command:

  ```
  mike@juniper1> file show <filename>
  ```

The File Copy Command

You can manually archive files with the `file copy` command which uses the same syntax as the standard UNIX `cp` command:

```
file copy /target-directory/target-filename /destination-
directory/destination-filename
```

For example, to copy the current active configuration file (/config/juniper.conf.gz) as backup.gz to the device's /var/home/user directory:

```
mike@juniper1> file copy /config/juniper.conf.gz /var/home/
user/backup.gz
```

BEST PRACTICE   Create a rescue configuration of a known working configuration. If the active configuration is corrupted, the device will automatically load the file named `rescue.gz` in the /config directory as the active configuration.

BEST PRACTICE   After copying the configuration file to a new location, always rename it so that you don't accidentally overwrite it later when copying an updated version of the file.

The same command lets you move the configuration file from the server back to the device's home directory:

```
mike@juniper1> file copy /config/juniper.conf.gz /var/home/
user/juniper.conf.gz-20090123
```

The File List Command

Use the `file list` command to verify that the file arrived in your home directory:

```
mike@juniper1> file list

/var/home/user/:
.ssh/
juniper.conf.gz-20090123
```

NOTE Chapter 4 includes the steps for loading the file as the active (running) configuration for the device.

Managing the Operating System Software

Operational mode provides commands for managing the operating system software, including upgrading and rebooting the device, as well as for restarting and resetting individual processes. JUNOS Software is a modular operating system whereby independent processes run in their own protected memory space. As such, these processes (called daemons) can be independently managed.

The Restart Command

You can restart most JUNOS Software processes from the operational mode (with a few daemons requiring that you must exit to a shell). Use `restart` when you need to stop and then restart individual operating system daemons.

ALERT! Although each process is fully independent, take special care when using the `restart` command. A restart of the SNMP process is only disruptive to SNMP, but a restart of routing could have drastic consequences in your network!

TIP To restart a specific routing protocol, such as OSPF, you can deactivate and then reactivate it in configuration mode. When a problem exists with only one protocol, this is a better approach than restarting the entire routing daemon of JUNOS, which would affect all the routing protocols.

The Request Command

The `request` commands perform system-wide functions such as rebooting, upgrading, and shutting down the device. This command group also provides the ability to online, offline, and restart individual components without having to reboot the entire device:

```
mike@juniper1> request chassis fpc slot 0 restart
Restart initiated, use "show chassis fpc" to verify
user@host> show chassis fpc
                   Temp    CPU Utilization (%)     Memory Utilization (%)
Slot State         (C)     Total Interrupt         DRAM (MB) Heap  Buffer
0 Starting         32        0        0                0        0      0
1 Online           30        0        0                8       11     14
2 Empty
3 Empty
```

MORE? The online technical documentation *JUNOS Software Installation and Upgrade Guide* provides details about upgrading the software version of your device. You can download the current package from the software download page at www.juniper.net/support. Downloading new software requires a current service contract and login account.

MORE? To learn more about operational mode commands, see the online *JUNOS Software CLI User Guide* and *JUNOS System Basics and Services Command Reference* at www.juniper.net/techpubs/.

Chapter 4

Discovering Configuration Mode

In configuration mode, as the name implies, you define the *configuration* of your device. This includes configuring the management console with its network settings, setting up user accounts for access to the device, specifying the security measures used to protect the device and the network, and setting up routing and switching protocols. Each statement configures different functions of the device, specifying its particular properties in your network.

Introducing the Configuration Process

The JUNOS CLI is thoughtfully designed to consider configuration as a process. Thus, safeguards have been introduced that allow you to set up and check a new configuration before it goes live. For example, JUNOS Software captures all changes in a *candidate* configuration that when completed can be committed, and only then can become the active configuration file.

This approach substantially contrasts with other systems that use line-by-line entry and instant activation of configuration changes. Have you ever had to make line-by-line changes in other systems, knowing that you were creating intermediate risks, such as removing a firewall on an interface? Perhaps you have entered a single-line change that created unwanted or unexpected results that you could not easily revert.

The JUNOS CLI protects you from these configuration headaches. With the help of early customers, the Juniper engineers purposefully designed a multi-stage configuration process. This process provides various methods of averting difficulties caused by unexpected mistakes and other common challenges in device configuration.

Figure 4.1 outlines the three basic steps to configure a device run by JUNOS Software. Here are the steps identified in the figure.

Where is the candidate?
Although it is easy to think of configurations as files, actually there is no file associated with the *candidate* configuration. The configuration is held in system memory.

1. **Make changes to the candidate configuration**

   The candidate configuration is a copy of the active configuration. You can enter configuration changes to the candidate through the CLI through cut and paste, load or merge a text file with the updated configuration changes, or enter the changes by hand through the CLI. After making all your candidate changes, you can review your work, including comparing the candidate to the active (running) file.

2. **Commit your changes.**
    To move the candidate to become the active configuration, enter
    the commit or commit confirmed commands. Before finalizing the
    changeover, the software checks for certain statements within the
    candidate and performs other context validations. If the device
    includes preloaded commit scripts, these scripts will also check
    and possibly correct errors within the candidate configuration.

3. **Candidate becomes active.**
    The candidate becomes active after passing through all the valida-
    tion checks. The candidate configuration becomes the active
    configuration, saved as /config/juniper.conf.gz. The device
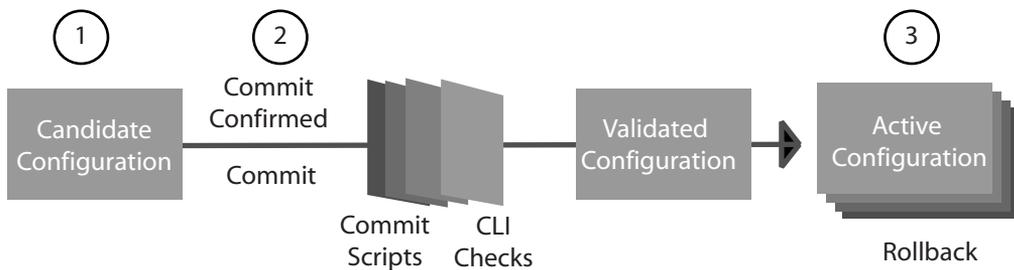   renames the previous juniper.conf.gz file to juniper.conf.1.gz.

Figure 4.1   **JUNOS CLI Configuration Process Overview**

NOTE   The JUNOS device saves up to 49 previous active configurations. You
        can roll back to any one of these backup configurations by issuing the
        rollback [0 - 49] command, discussed later in this chapter.

Entering the Configuration Mode

In devices where different user accounts can make configuration
changes, the flexibility to manage who is making changes and when
they make them is essential. JUNOS Software thus offers three options
for entering configuration mode:

■ Standard: allows any number of users to edit the candidate
  configuration simultaneously, and changes made by a single user
  are visibly shared so that they can be seen by all users.

■ Exclusive: locks all other users out of configuration mode until the
  exclusive user closes the exclusive state.

■ Private: provides a private configuration, whereby the device keeps a separate candidate copy holding only the changes by the private user.

The Configure Command

To enter the standard configuration mode, issue the `configure` command:

```
mike@juniper1> configure
Entering configuration mode
```

ALERT! When you exit from the standard configuration mode, all the uncommitted changes you have made during your session remain in the candidate, unless you explicitly delete them or issue a `rollback 0` command to reload the active configuration as the candidate. The users do get warning messages when logging in and out:

```
mike@juniper1> configure
Entering configuration mode
The configuration has been changed but not committed

mike@juniper1# exit
The configuration has been changed but not committed
Exit with uncommitted changes? [yes,no]
```

The Configure Exclusive Command

To lock the candidate configuration from other users, add the `exclusive` switch to the `configure` command. In configure exclusive mode, the device discards all noncommitted changes to the configuration once you exit the session.

```
mike@juniper1> configure exclusive
warning: uncommitted changes will be discarded on exit
Entering configuration mode
```

The Configure Private Command

You can create your own private candidate configuration by adding the `private` switch to the `configure` command:

```
mike@juniper1> configure private
warning:uncommitted changes will be discarded on exit
Entering configuration mode
```

When a private user commits changes, JUNOS Software integrates only the candidate changes made by the private user into the active (running) configuration. The software does not implement any pieces of the candidate configuration changed by others. This means that several users can use `configure private` to make nonconflicting changes to the active configuration at the same time. If a private user issues a `rollback 0` command, the device discards only that user's changes in candidate configuration.

NOTE   If a user creates a private configuration session, other users can log in as usual or in their own private session. When a person is already logged in, other users are warned that another person is currently modifying the configuration.

BEST PRACTICE   Use `configure exclusive` or `configure private` whenever multiple user accounts can make changes to the configuration. This best practice protects everyone from inadvertent errors. For instance, the Juniper engineers heard about one event when an administrator accidentally typed the command `delete interfaces`. The administrator recognized the mistake but instead of removing the statement, he simply exited configuration mode. Later, when another user logged in and committed the configuration, all the device's interfaces were deleted! Fortunately, JUNOS Software made it possible to roll back to a previous configuration.

Understanding Configuration Mode Basics

Configuration mode offers several options to view and navigate the candidate configuration.

Viewing the Candidate Configuration

The `show` command displays the candidate configuration of the device. When this command is entered from the top of the configuration hierarchy, the CLI displays the entire candidate configuration. The following example provides an abbreviated listing for a configured device:

```
[edit]
mike@juniper1# show
version "9.2R1.3";
```

```
groups
{
   re0 {
     system {
       host-name juniper1;
     }
   }
}
<snip>
```

If you haven't made any configuration changes, then the candidate configuration is the same as the active (running) configuration of the device.

Deeper in the hierarchy, the show command displays the configuration from the current configuration hierarchy level and below:

```
[edit interfaces ge-5/0/0]
mike@juniper1# show
gigether-options {
    flow-control;
    auto-negotiation;
}
unit 0 {
    family inet {
        address 1.2.3.4/28;
    }
}
```

NOTE   You may have noticed that the configuration mode uses the show command from operational mode. The commands of each mode are independent of each other, and so the show command represents different actions in each mode.

Navigating the Configuration

Although you can edit the configuration from the root of the hierarchy, it is often easier to navigate to the area within the configuration you are changing before adding and removing commands. For example, if you were planning to add new services to the configuration, you could issue the following series of set commands:

```
[edit]
mike@juniper1# set system services finger
mike@juniper1# set system services ftp
mike@juniper1# set system services ssh
mike@juniper1# set system services telnet
```

However, it is easier to navigate to the system services directory and then issue the following commands:

```
[edit system services]
mike@juniper1# set finger
mike@juniper1# set ftp
mike@juniper1# set ssh
mike@juniper1# set telnet
```

In either case, when you have edited the configuration, the following lines are added to the candidate configuration:

```
[edit]
system {
    services {
        finger;
        ftp;
        ssh;
        telnet;
    }
}
```

The CLI provides four commands for navigation in configuration mode: edit, up, top, and exit.

The Edit Command

Use the edit command to jump to a specific location within the candidate configuration. The configuration mode banner changes to indicate your new location in the hierarchy:

```
[edit]
mike@juniper1# edit system services

[edit system services]
mike@juniper1#
```

You do not have to issue the edit command from the top level directory. For example, to navigate to the system syslog host log hierarchy, you could issue the following command from the top level of the hierarchy:

```
[edit]
mike@juniper1# edit system syslog host log

[edit system syslog host log]
mike@juniper1#
```

You could also navigate to the same hierarchy by issuing the following succession of `edit` commands:

```
[edit]
mike@juniper1# edit system

[edit system]
mike@juniper1# edit syslog

[edit system syslog]
mike@juniper1# edit host log

[edit system syslog host log]
mike@juniper1#
```

When issuing the `edit` command from the hierarchy, issue the relative path based on your location in the hierarchy.

NOTE The `edit` command functions like the UNIX change directory (`cd`) command, moving you to an exact location in the hierarchy tree.

If you navigate to a hierarchy location that doesn't exist in your configuration yet, the CLI will create the hierarchy level. However, explicitly adding hierarchy levels using the `set` command (discussed below) helps you to know exactly what you have created.

The Up Command

The `up` command allows you to move up levels in the hierarchy. By default, you move one level. You can add a number after the command to specify how many levels to move up.

```
[edit interfaces fe-1/3/1 unit 0 family inet address 10.0.10.1]
mike@juniper1# up

[edit interfaces fe-1/3/1 unit 0 family inet]
mike@juniper1#
```

In the above example, `interfaces`, `fe-1/3/1`, `unit 0`, `family inet`, and `address 10.0.10.1` each represent one level within the hierarchy as shown below from the top of the configuration hierarchy:

```
[edit]
interfaces {
    fe-1/3/1 {
        unit 0 {
            family inet {
                address 10.0.10.1;
            }
        }
    }
}
```

The Top Command

The `top` command allows you to move to the first hierarchy level.

The Exit Command

The `exit` command returns you to the hierarchy location before the last `edit` command. If you issue this command from the top level of the configuration hierarchy, you exit configuration mode.

SHORTCUT You can combine navigation commands together to move through the hierarchy. For example, you can use `top` and `edit` together to move quickly to a different part of the configuration hierarchy:

```
[edit protocols ospf area]
mike@juniper1# top edit system login

[edit system login]
mike@juniper1#
```

Use `top` with `show` to display a portion of the configuration from another section of the hierarchy:

```
[edit protocols ospf area]
mike@juniper1# top show system services
web-management {
    http {
        port 8080;
    }
}
```

Try it Yourself: Navigating the Hierarchy

Go to the following configuration hierarchy levels using the `edit`, `up`, `top`, and `exit` commands:

```
[edit interfaces]
[edit interfaces fe-0/0/0]
[edit]
[edit protocols bgp]
[edit protocols ldp interface fe-0/0/0.0]
[edit system]
[edit system services telnet]
[edit system syslog archive]
[edit system syslog]
[edit snmp v3]
[edit snmp v3 usm local-engine]
[edit snmp v3 target-parameters sample-parameters parameters]
```

Editing the Configuration

Create or change the candidate configuration by entering a series of commands, including commands to add and remove configuration statements.

The Set Command

The `set` command inserts a statement and values into the candidate configuration. For example, if you want to add the FTP service to your device, from the top of the hierarchy you issue the following `set` command:

```
[edit]
mike@juniper1# set system services ftp
```

The following lines will be added to the configuration file:

```
system {
    services {
        ftp;
    }
}
```

To learn about using set commands to configure your device, download additional titles of the JUNOS Software Fundamentals Series Day One booklets as they become available at www.juniper.net/dayone/.

You also use the `set` command to add statement values when required. For example, to set the device name to `juniper1`, you enter the following `set` command:

```
[edit]
mike@devicename # set system host-name juniper1
```

The following lines will be added to the configuration file:

```
system {
    host-name juniper1;
}
```

The Delete Command

The `delete` command removes statements from your candidate configuration. Deleting a statement effectively returns the affected device, protocol, or service to an unconfigured state. Deleting a container statement removes everything under that level of the hierarchy.

ALERT!   The `delete` command removes all subordinate statements and identifiers. For example, the following simple line would remove all the protocol configuration data in your candidate:

```
[edit]
mike@juniper1# delete protocols
```

BEST PRACTICE   Know where you are in the hierarchy and everything that your command will remove when you issue a `delete` statement! By always checking the `[edit]` banner to determine your current hierarchy location, you can be sure your command affects only the portion of the configuration that you want to change.

If a configuration statement is empty after you delete the configuration element(s), the CLI removes that configuration statement from the candidate configuration.

Try it Yourself: Setting and Deleting Configuration Commands

Follow these steps to set up and then delete a serial interface with the IP address 10.210.1.0/24.

1. Enter the following `set` command at the top level of the configuration hierarchy:

```
[edit]
mike@juniper1# set interfaces se-1/0/0 unit 0 family inet address 10.210.1.0/24
```

2. Use the `show` command to verify that the se interface was added to the configuration (In the following example, only the added statements are shown; your configuration file should have more data than shown):

```
[edit]
mike@juniper1# show

interfaces {
    se-1/0/0 {
        unit 0 {
                family inet {
                    address 10.210.1.0/24;}
                }
            }
        }
}
```

3. Delete the interface using the following `delete` command:

```
[edit]
mike@juniper1# delete interfaces se-1/0/0
```

4. Use the `show` command to verify that the se interface is now removed from your configuration file.

MORE?   When you need to remove large common pieces of the configuration from the device, wildcards can save you time. The device can search through the entire candidate configuration looking for a string and delete every line that contains that string. To learn more about wildcards, go to the online *JUNOS Software CLI User Guide* at www.juniper.net/techpubs/ and refer to the section on Advanced Features.

The Annotate Command

The JUNOS CLI lets you leave comments about the configuration as a part of its listing. The comments can be quite handy when you or other team members are trying to troubleshoot a problem or need to make configuration changes. Issue the `annotate` command followed by your note when you want to include comments:

```
[edit]
mike@juniper1# annotate system  this device is for training new
JUNOS users
```

When you add comments in configuration mode, they are associated with a statement at the current level. Each statement can have one single-line comment associated with it. To delete a comment, use the `annotate` command with an empty string:

```
[edit]
mike@juniper1# annotate system ""
```

Committing the Candidate Configuration

The JUNOS CLI provides multiple features that help users to catch and correct typos, omissions, and other errors before they become a problem. In addition to candidate configurations, these features include providing file comparisons, checking candidate syntax and context, enabling fast rollback, and restoring working configurations on systems that become isolated after activation of a new configuration.

Figure 4.2 provides a detailed view of the file management of the device configuration. The active (running) configuration is the operational file of the device. It is also the configuration that the device loads during a boot sequence. The candidate configuration is the working copy storing configuration updates. The `commit` commands cause the following transitions by the device (for candidates which pass the validation checks):

1. Copies the candidate configuration to the active configuration. At this point, the active and the candidate configurations are identical.

2. Decrements all rollback configuration files by one and saves the active configuration as rollback 0.
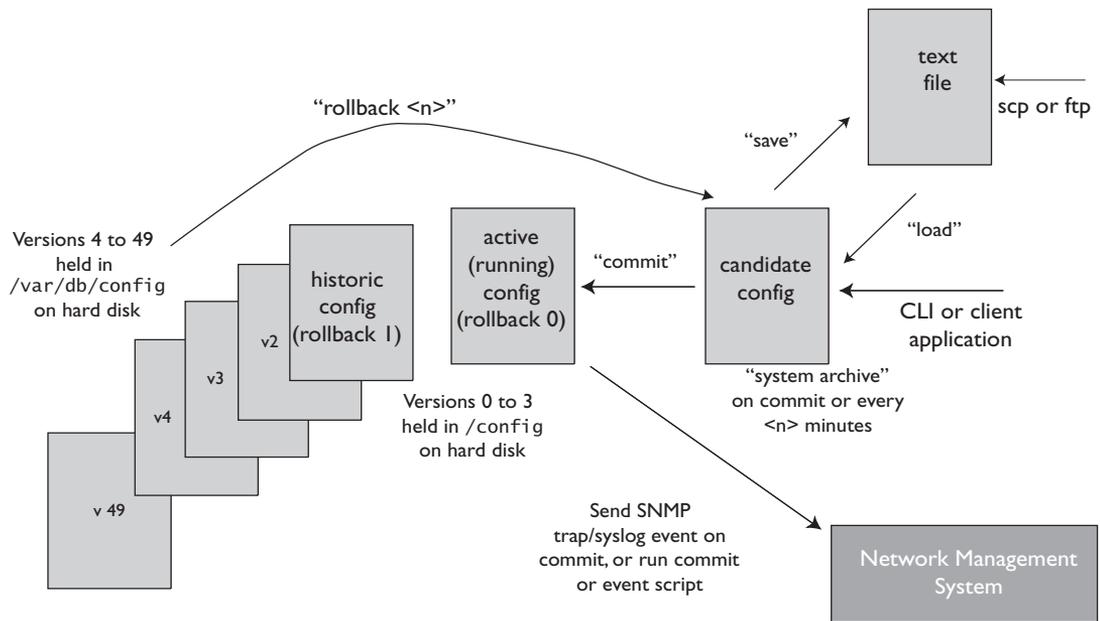
Figure 4.2   **JUNOS Configuration File Management**

The active (running) configuration file and the last three rollback configuration files are saved in the /config directory. The device saves the remainder of the archived configuration files in the /var/db/config directory.

NOTE   The active configuration file is named juniper.conf.gz, and the rollback configuration files are named from juniper.conf.1.gz to juniper.conf.49.gz (providing an archive of 50 active configurations).

The Compare Command

Configuration mode conveniently provides a way to display the configured differences between two configurations with the show | compare command.

The following example modifies a candidate configuration by enabling Telnet access and removing SSH and J-Web access:

```
[edit system]
mike@juniper1# set services telnet

[edit system]
mike@juniper1# delete services web-management

[edit system]
mike@juniper1# delete services ssh
```

Now display the resulting changes in the candidate compared to the active configuration:

```
[edit system services]
mike@juniper1# show | compare
- ssh;
+ telnet;
- web-management {
-     http {
-         port 8080;
-     }
- }
```

The command interface indicates new lines in the candidate with a plus (+) sign and those removed with a minus (-) sign.

SHORT CUT The operational mode command show configuration displays the current active (running) configuration. You can perform this command in configuration mode by adding the keyword run:

```
[edit]
mike@juniper1# run show configuration
```

The Commit Check Command

The CLI also provides a command to check that the system can process your candidate configuration. The commit check command validates the logic and completeness of the candidate without activating any

changes. These are the same validations that run when you commit a candidate. If the system finds a problem in the candidate configuration, it lets you know:

```
[edit]
mike@juniper1# commit check
[edit interfaces lo0 unit 0 family inet]
  'address 192.168.69.1/24'
    Loopback addresses' prefix must be 32 bits
error: configuration check-out failed
```

BEST PRACTICE Before activating a candidate as the running configuration, always check your work. Use the show | compare command to ensure all of the expected configuration elements and parameters are a part of the candidate. Enter the commit check command to have the system validate your candidate configuration without activating the changes.

The Commit Command

The candidate file is only the proposed configuration, and your device does not use any of this configuration until you issue a commit command. After you have entered all desired changes and you have double-checked your work, you are ready to activate your candidate as the active (running) configuration.

To activate the candidate configuration, enter the commit command:

```
[edit]
mike@juniper1# commit
commit complete
```

Before actually activating the candidate configuration, JUNOS Software checks basic syntax and semantics. For example, the software makes sure that a policy has been defined before it is referenced. If any syntax or semantic problems are found, the commit command returns an error:

```
[edit]
mike@juniper1# commit
error: Policy error: Policy my-policy referenced but not defined
error: BGP: export list not applied
error: configuration check-out failed
```

You must fix all mistakes before the candidate (or any part of the candidate) can become active.

The commit complete message tells you that the new configuration is up and running on the device:

```
[edit]
mike@juniper1# commit
commit complete
```

ALERT! By default, if more than one user is modifying the configuration, committing the configuration saves and activates the changes of all users (unless a user is in configure private mode; see pages 40–41).

The Commit Confirmed Command

Are you among those of us who have made the mistake of adding security to a remote box, only to discover the new firewall locked you out of the very interface that you were using to access the device? Do you have a story about the time you accidentally isolated a remote box and then had to jump in the car and drive for four hours in the middle of the night just to reset it? The commit confirmed command can prevent costly configuration mistakes by automatically rolling back problematic configurations.

The commit confirmed command commits a candidate configuration for 10 minutes. If you don't then follow up with a second commit command, the device automatically rolls back to the previous configuration. You can use the commit confirmed command anytime you want a safety net against potential configuration problems:

```
[edit]
mike@juniper1# commit confirmed
commit confirmed will be automatically rolled back in 10 minutes
unless confirmed
commit complete
```

If everything looks good, then you need to commit the new configuration a second time for the configuration to become permanent:

```
[edit]
mike@juniper1# commit
commit complete
```

If you do not confirm the configuration by entering a second commit command, the CLI will roll back the device to the previous active configuration at the end of the 10 minutes. In this way, if you have accidently isolated the device, you simply need to wait for the rollback instead of agonizing over how you are going to otherwise undo your mistake:

```
Broadcast Message from root@juniper1
        (no tty) at 08:10:17 UTC
Commit was not confirmed; automatic rollback complete.
```

After the device rolls back, check for errors in the candidate configuration, and then try the commit command again.

You can alter the time that the device waits before rolling back by adding a wait-time (in minutes) to the command:

```
[edit]
mike@juniper1# commit confirmed 2
commit confirm will be automatically rolled back in 2 minutes
unless confirmed commit complete
```

BEST PRACTICE   When you are configuring remote devices, always use the commit confirmed command to activate your candidate configuration. Even the most experienced JUNOS users want the insurance policy it provides to their work, and many have a story to tell about the day it saved them from their own mistake and lots of extra work.

Automating Everyday Tasks

You can optimize your work by adding automated scripting to your repertoire. JUNOScript lets you automate, in your own way, both configuration and operational CLI commands. Using automated scripting can not only save you time, but can also reduce downtime by preventing configuration errors and speeding problem resolution and restoration.

Commit scripts enable customized checks of configurations to ensure that they are in compliance with your network standards and policies. *Op* and *event scripts* can receive command output, inspect it, and determine the next appropriate action—repeating the process until a problem's source has been found.

MORE?   For instructions on implementing scripts, see the *Configuration and Diagnostic Automation Guide* at www.juniper.net/techpubs/.

Rolling Back the Configuration

Whenever you commit the candidate configuration as the new active configuration, JUNOS Software automatically saves a copy of the replaced active file. As you store each newly replaced configuration, all the prior configuration files move back one version number further in the configuration archive. Each device can store up to 49 of the most recently active versions along with the current active configuration (also known as `rollback 0`).

You can access this configuration archive using the `rollback` command, including the number of versions you want to go back. Return to the most recent previous configuration file using the `rollback 1` command:

```
[edit]
mike@juniper1# rollback 1
load complete
```

The `rollback` command loads the requested archive as the candidate file. If you want to use it immediately, first make sure it's what you want by using the `show` command, and then activate it with the `commit` command:

The `rollback rescue` command loads the rescue configuration file (see the BEST PRACTICE tip on page 34 for more information).

```
[edit]
mike@juniper1# show
<snip>

[edit]
mike@juniper1# commit
commit complete
```

This automatic backup mechanism lets you return quickly to a previous configuration for immediate use or for fast updates.

ALERT!   Don't forget it's necessary to commit the candidate file to actually activate the selected rollback file as the running configuration.

TIP   If you aren't sure what differences exist between the active (running) configuration and a rollback file, investigate with the `show | compare` command:

```
[edit interfaces]
mike@juniper1# show | compare rollback 2

[edit interfaces]
```

```
-   fe-3/0/1 {
-       vlan-tagging;
-       unit 240 {
-           vlan-id 240;
-           family inet {
-               address 10.14.250.1/28;
-               address 10.14.250.17/28 {
-                   preferred;
-               }
-               address 10.14.250.33/28;
-               address 10.14.250.49/28;
-               address 10.14.250.65/28;
-           }
-       }
- }
```

TIP   Use the question mark with the `rollback` command to list the full archive:

```
[edit]
mike@juniper1# rollback ?
Possible completions:
<[Enter]>            Execute this command
0                    2009-01-31 04:34:56 UTC by mike via cli
1                    2009-01-31 04:30:03 UTC by mike via cli
2                    2009-01-30 06:23:44 UTC by mike via cli
<snip>
48                   2008-11-03 08:00:03 UTC by mike via cli
49                   2008-11-03 07:45:21 UTC by mike via cli
  |                   Pipe through a command
```

TIP   To reset the candidate configuration to the currently active configuration use the `rollback` (or `rollback 0`) command.

Preparing System Changes in Advance

The JUNOS CLI provides options for making system changes in advance. These options can remain inactive parts of your configuration until needed or become scheduled elements with a set activation time.

Preconfiguration of New Hardware

Unlike other systems, JUNOS Software lets you prepare for an installation before actually installing the hardware. The software simply ignores any parts of the running configuration that are irrelevant to

the existing hardware installation. Whenever the hardware becomes available, the newly added section of the configuration then becomes active.

The option to set up a configuration before hardware installation is quite useful, especially when the person installing the hardware is different from the person configuring the device, a common occurrence for remote boxes. Here is a configuration for fe-3/0/0, for example, which could be installed tomorrow:

```
[edit]
mike@juniper1# edit interfaces fe-3/0/0 unit 0

[edit interfaces fe-3/0/0 unit 0]
mike@juniper1# set family inet address 192.168.1.254/24

[edit interfaces fe-3/0/0 unit 0]
mike@juniper1# commit
commit complete
```

The Commit At Command

Sometimes you want to prepare configuration changes for activation at a specific time, such as during a maintenance window. The commit at command provides this option:

```
[edit]
mike@juniper1# commit at 02:00:00
commit check succeeds
commit will be executed at 2009-02-02 02:00:00 UTC
Exiting configuration mode
mike@juniper1>
```

To display any pending commit operations (and the commit history), enter the show system commit command. You can cancel a pending commit operation with the clear system commit command.

```
mike@juniper1>clear system commit
  Pending commit cleared
```

The Deactivate Command

You can also make configuration changes and mark them as inactive until you are ready to use them. The device ignores these portions of the configuration as though they were not even defined. In this exam-

ple, a new BGP neighbor at 192.168.1.1 is configured but left deacti-
vated until the session is ready to be introduced:

```
[edit]
mike@juniper1# edit protocols bgp group internal

[edit protocols bgp group internal]
mike@juniper1# set neighbor 192.168.1.1

[edit protocols bgp group internal]
mike@juniper1# deactivate neighbor 192.168.1.1

[edit protocols bgp group internal]
mike@juniper1# show
type internal;
local-address 10.14.243.255;
export [ nhs accept-aggregates ];
neighbor 10.14.243.254;
inactive: neighbor 192.168.1.1;

[edit protocols bgp group internal]
mike@juniper1# commit
commit complete
```

When you're ready to make the change, you just activate and commit
that portion of the configuration, and the device will begin using it:

```
[edit protocols bgp group internal]
mike@juniper1# activate neighbor 192.168.1.1

[edit protocols bgp group internal]
mike@juniper1# commit
commit complete
```

You can deactivate any portion of the configuration hierarchy, and the
device ignores everything underneath it. For example, you can deacti-
vate the entire group of BGP neighbors called internal:

```
[edit protocols bgp group internal]
mike@juniper1# up

[edit protocols bgp]
mike@juniper1# deactivate group internal
```

Using Configuration Shortcuts

A typical configuration includes many similar elements named and defined by the user, such as interface names, policy statements, and firewall filters. The JUNOS CLI includes commands to duplicate and quickly change the configurations of these user-defined elements.

The Copy Command

The copy command duplicates a configuration statement along with all the subordinate statements configured underneath it. In using the command, you copy the configuration associated with one user-defined element to a new, similarly configured element. You then modify that second element with any needed changes.

The following sample configuration shows a configured serial interface se-0/0/2:

```
[edit interfaces]
mike@juniper1# show
se-0/0/2 {
  serial-options {
    clocking-mode dce;
    clock-rate 125.0khz;
  }
  unit 0 {
    point-to-point;
    family inet {
      address 10.0.22.1/24;
    }
  }
}
```

You can use the copy command to set up a new interface se-0/0/1. The duplicated interface has exactly the same parameters as the original. You can then make any needed changes in the configuration of the newly created interface se-0/0/1; for example, you can change its address:

```
[edit interfaces]
mike@juniper1# copy se-0/0/2 to se-0/0/1

[edit interfaces]
mike@juniper1# delete se-0/0/1 unit 0 family inet address
10.0.22.1/24
```

You've now deleted the copied address. Replace it with the correct address for the new interface:

```
[edit interfaces]
mike@juniper1# set se-0/0/1 unit 0 family inet address
10.0.36.2/24
```

Very quickly, the new se-0/0/1 interface has been created, keeping most of the same properties as the se-0/0/2 interface:

```
[edit interfaces]
mike@juniper1# show
se-0/0/1 {
  serial-options {
    clocking-mode dce;
    clock-rate 125.0khz;
  }
  unit 0 {
    point-to-point;
    family inet {
      address 10.0.36.2/24;
    }
  }
}

se-0/0/2 {
  serial-options {
    clocking-mode dce;
    clock-rate 125.0khz;
  }
  unit 0 {
    point-to-point;
    family inet {
      address 10.0.22.1/24;
    }
  }
}
```

ALERT!   Before commiting the candidate file, be sure to check your edits when changing the configuration with the copy command. Check that you made all the modifications needed in all the duplicated statements.

Remember, the configuration might not be valid immediately after you have copied a portion of the configuration. You must check the validity of the new configuration and, if necessary, make modifications for the configuration to be valid.

The Rename Command

The `rename` command is a convenient shortcut when you need to alter the value of a user-defined variable—such as policy names, filter names, or IP addresses—or to change the name of a user-defined element.

In the next example, the address of the Fast Ethernet fe-4/0/2 interface has been incorrectly set to 10.73.24.103/24.

```
[edit interfaces]
mike@juniper1# show

fe-4/0/2 {
  unit 0 {
    family inet {
      address 10.73.24.103/24;
    }
  }
}
```

Use the `rename` command to change the value to 10.73.24.143/24:

```
[edit interfaces]
mike@juniper1# rename fe-4/0/2 unit 0 family inet address
10.73.24.103/24 to address 10.73.24.143/24
```

Check to see that the change is quickly completed:

```
[edit interfaces]
mike@juniper1# show

fe-4/0/2 {
  unit 0 {
    family inet {
      address 10.73.24.143/24;
    }
  }
}
```

NOTE Alternatively, instead of using `rename`, you can use the `delete` command to remove the statement and then use the `set` command to add the new value.

Switching Ports: A Useful Configuration Trick

How many times have you had to temporarily move a connection to another port just to test it? In JUNOS Software, the process is simple. Follow this example, as the configuration is moved from fe-2/0/1 to fe-2/0/0.

Begin by looking at the existing interface configuration:

```
[edit]
mike@juniper1# show interfaces
fe-0/0/0 {
    description "MGMT INTERFACE - DO NOT DELETE";
    unit 0 {
        family inet {
            address 10.210.9.177/28;
        }
    }
}
fe-2/0/1 {
    vlan-tagging;
    unit 240 {
        vlan-id 240;
        family inet {
            address 10.14.243.238/28;
        }
    }
}
```

The following moves the port in the candidate file:

```
[edit]
mike@juniper1# rename interfaces fe-2/0/1 to fe-2/0/0
```

The candidate configuration now shows this move:

```
[edit]
mike@juniper.net# show interfaces
fe-0/0/0 {
    description "MGMT INTERFACE - DO NOT DELETE";
    unit 0 {
        family inet {
            address 10.210.9.177/28;
        }
    }
}
fe-2/0/0 {
```

```
            vlan-tagging;
            unit 240 {
                vlan-id 240;
                family inet {
                    address 10.14.243.238/28;
                }
            }
        }
```

MORE? To learn more about configuration mode commands, see the online *JUNOS Software CLI User Guide* at www.juniper.net/techpubs/.

What to Do Next & Where to Go ...

www.juniper.net/junos

> Everything you need for JUNOS Software adoption and education.

forums.juniper.net/jnet

> The Juniper-sponsored J-Net Communities forum is dedicated to sharing information, best practices, and questions about Juniper products, technologies, and solutions. Register to participate in this free forum.

www.juniper.net/techpubs

> All product documentation developed by Juniper Networks is freely accessible at this site. Find what you need to know about JUNOS Software under each product line.

www.juniper.net/books

> Juniper works with multiple book publishers to author and publish technical books on topics essential to network administrators. Check out their ever-expanding list of newly published books.

www.juniper.net/training/fasttrack

> Take courses online, on location, or at one of the partner training centers around the world. The Juniper Network Technical Certification Program (JNTCP) allows you to earn certifications by demonstrating competence in configuration and troubleshooting of Juniper products. If you want the fast track to earning your certifications in enterprise routing, switching, or security, use the available online courses, student guides, and lab guides.

Do you have a comment about this Day One booklet? Drop us an email at dayone@juniper.net.

Command Reference

(Summary of commands discussed in this Day One booklet.)

Configuration Mode Commands

activate Activate a portion of the configuration.

annotate Leave comments about a configuration statement.

commit Commit candidate set of changes.

commit at Commit candidate at a set time.

commit check Validate the candidate configuration without activating any changes.

commit confirmed Automate rollback if the user does not follow the command with a confirmation.

compare Display the differences between the two configurations.

copy Copy a statement.

deactivate Mark portions of the configuration as inactive.

delete Remove a configuration statement(s) or identifier.

edit Move to the designated hierarchy level.

exit Exit this level of the configuration hierarchy. At the top level, exit configuration mode.

exit configuration-mode Exit configuration mode.

help Get onboard help.

pipe Take output from one command and use it as input to another command or redirect the output to a file.

rename Assign a new name to a configuration or identifier.

rollback Restore the candidate file to a previous committed configuration.

run Run an operational mode command.

set Create a statement hierarchy and set identifier values.

show Display the candidate configuration.

top Move to the first hierarchy level.

up Move up one level in the hierarchy.

Operational Mode Commands

clear Remove system information.

configure Enter configuration mode.

configure exclusive Get an exclusive lock on the candidate so others can't edit it.

configure private Give the user their own copy of the candidate configuration.

exit Exit operational mode.

file copy Create and archive files.

file list List files and directories on the device.

file show View the file contents.

help Get onboard help.

monitor Show real-time debugging information.

ping Send a message to another host to verify connectivity.

pipe Take output from one command and use it as input to another command or redirect the output to a file.

request Install new software versions, reboot, shut down.

restart Restart individual operating system daemons.

set Establish system properties.

show Show system information.

ssh Start secure shell on another host.

start shell Log in to the C shell interface.

telnet Open a terminal connection to another device or host on the network.

traceroute Record and display every IP packet hop from one location to another.