Junos® OS Fundamentals **Series**

DAY ONE: EXPLORING THE JUNOS CLI, SECOND EDITION

The **two** most popular books on the Junos OS are now combined, revised, and updated into **one book!**

It's day one and you have a job to do, so start using the Junos CLI. It's fast, it's easy, and you're just a few hours away from modifying, saving, and loading configuration files onto your device.

By Walter Goralski, Sean Clarke, and Ian Jarrett

DAY ONE: EXPLORING THE JUNOS CLI, SECOND EDITION

The Junos OS command-line interface (CLI) includes dozens of shortcuts to get things done in your network. You'll spend much less time pounding away on your keyboard once you master these commands, and soon, with just a little effort, you'll learn why so many people say that the Junos OS saves time (often lots of it), reduces repetitive tasks, and helps to avoid costly mistakes.

Day One: Exploring the Junos CLI, Second Edition is for beginning users of devices running the Junos OS, or as a refresher course when it's time to scale Juniper technology. It not only lays the foundation for learning the Junos OS, but also facilitates understanding of the more advanced Junos OS books that populate the *Day One* library.

This Second Edition combines two previous best-selling Day One books – *Day One: Exploring the Junos CLI* and *Day One: Configuring Junos Basics* – into a single updated and revised Junos OS book that gets you started and then helps you get things done.

IT'S DAY ONE AND YOU HAVE A JOB TO DO, SO LEARN 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 OS commit model.
- Prepare system changes in advance.
- Use the shortcuts and tips of experienced users and avoid common problems.

Juniper Networks Books are singularly focused on network productivity and efficiency. Peruse the complete library at www.juniper.net/books.

Published by Juniper Networks Books

ISBN 978-1941441237

51600

9 781941 441237

JUNIPER NETWORKS

Junos® OS Fundamentals Series

Day One: Exploring the Junos OS CLI, Second Edition

By Walter Goralski, Sean Clarke, and Ian Jarrett

JUNIPer NETWORKS

Published by Juniper Networks Books
Authors: Walter Goralski, Sean Clarke, Ian Jarrett
Technical Reviewers: Nighat Ara, Rashmi Nadig, Kenneth Pacunas, Bryan Phillips, Chaitra Satish
Editor in Chief: Patrick Ames
Copyeditor and Proofer: Nancy Koerbel
Illustrator: Karen Joice
J-Net Community Manager: Julie Wider

ISBN: 978-1-941441-23-7 (print)
Printed in the USA by Vervante Corporation.
ISBN: 978-1-941441-24-4 (ebook)

Version History: v1, November 2015
2 3 4 5 6 7 8 9 10

This book was originally published as two books, *Day One: Exploring the Junos CLI* and *Day One: Configuring Junos Basics*. It has been updated, revised, and technically reviewed to match current Junos OS operations.

This book is available in a variety of formats at: http://www.juniper.net/dayone.

Second Edition Author
Walter Goralski has been involved in networking and the Internet for many years. He has been an instructor, course developer, college professor, technical writer, and author of bestselling technical books and specializes in making new and complex technologies easy for everyone to understand. He has worked for Juniper Networks since 2000 and is now a member of Juniper's iLX Solutions Group.

First Edition Authors
Sean Clarke has over 15 years experience working for Juniper Networks, focusing on Service Provider and Data Center technologies. He is currently employed in the Proof of Concept lab, Amsterdam.

Ian Jarrett has over 20 years experience in the networking and telecommunications industry and has worked with the Junos OS since 1998. He is currently the Professional Services Theater Practice Lead for OSS and Automation with Juniper Networks in EMEA.

Second Edition Reviewers
Nighat Ara is a Network Test Engineer in the PDT team at Juniper Networks. She has over nine years of experience in test/validation, customer support, and technical instruction. She has a BS and MS in Electrical Engineering.

Rashmi Nadig is a recent college graduate and has been working as a Test Engineer in the Junos Kernel System test team at Juniper Networks for the past ten months, where she performs requirements-based testing and automation on the latest Junos OS features and networking advancements.

Kenneth Pacunas has been in the networking industry 20 plus years and his current responsibilities include Junos OS regression testing, debugging, and script fixing / modification. “This *Day One* book is a very good first step – just the right length and relevant content.”

Bryan Phillips has over twenty years experience in the networking and telecommunication industry, with the last ten years focused on the latest MPLS technologies. Bryan is currently employed by Juniper Networks, where he functions as a Test Engineer Staff in the Routing Business Unit.

Chaitra Satish is a Quality Assurance Engineer in the Junos Kernel SysTest team at Juniper Networks where she works on the latest cutting edge features. Chaitra also has experience providing customer support for the Juniper SRX Series security products.

Welcome to Day One

This book is part of a growing library of *Day One* books, produced and published by Juniper Networks Books.

Day One books were conceived to help you get just the information that you need on day one. The series covers Junos OS and Juniper Networks networking essentials with straightforward explanations, step-by-step instructions, and practical examples that are easy to follow.

The *Day One* library also includes a slightly larger and longer suite of *This Week* books, whose concepts and test bed examples are more similar to a weeklong seminar.

You can obtain either series, in multiple formats:

- Download a free PDF edition at http://www.juniper.net/dayone.
- Get the ebook edition for iPhones and iPads from the iTunes Store. Search for *Juniper Networks Books*.
- Get the ebook edition for any device that runs the Kindle app (Android, Kindle, iPad, PC, or Mac) by opening your device's Kindle app and going to the Kindle Store. Search for *Juniper Networks Books*.
- Purchase the paper edition at either Vervante Corporation (www.vervante.com) for between $12-$28, depending on page length.

Audience

This book is intended for network engineers who have just begun their career in network engineering using the Junos OS.

What You Need to Know Before Reading This Book

This book is intended for those readers who are new to the Junos OS CLI. Familiarity with other CLI-based operating systems is an advantage, but not a requirement. Other *Day One* books in the *Junos OS Fundamentals Series* can help you with device and configuration details: http://www.juniper.net/dayone.

NOTE Having access to a device running the Junos OS is useful as you follow along with the steps and configurations in this book's examples.

After Reading This Book You Will Be Able 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 OS commit model.
- Prepare system changes in advance.
- Use the shortcuts and tips of experienced users and avoid common problems.

Information Experience

This *Day One* book is singularly focused on networking fundamentals and it is highly recommended you read and review the Junos OS technical documentation in order to become fully acquainted with the initial configuration process of devices that run the Junos OS.

All Juniper technical documentation is located at http://www.juniper.net/documentation.

Chapter 1

Introducing the Junos OS CLI

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

If you've operated other networking devices, the Junos OS CLI should seem familiar, but you will also quickly notice that it includes some new and different commands. No need to fret. The Junos OS CLI offers a rich set of tools and safeguards that help you efficiently manage your network and maintain high uptime.

The command-line interface includes lots of shortcuts and commands for you to get help. Master them, 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 the Junos OS saves them time (often lots of it), reduces repetitive tasks, and helps them to avoid mistakes.

NOTE 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. Download the *J-Web User Interface Guide* at: http://www.juniper.net/documentation/en_US/junos15.1/information-products/pathway-pages/jweb/jweb.pdf.

Outlining the Command Modes

The first step in exploring the Junos OS CLI is understanding its two command modes:

- *Operational mode*: manages and monitors device operations. For example, you can monitor the status of the device interfaces, check chassis alarms, and upgrade and downgrade the device's operating system. Operational mode uses the > prompt.
- *Configuration mode*: configures the device and its interfaces. These include user access, interfaces, protocols, security services, and system hardware properties. Configuration mode uses the # prompt.

All commands are case-sensitive, so beware of the Caps Lock key. If you type a capital letter when the system is expecting a lower case letter, you will get a syntax error.

CLI Modes

The Junos OS 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 OS CLI preferred by users. By logically grouping activities, the Junos OS CLI provides a regular, consistent syntax helpful for knowing where you are, finding what you want, moving around the interface, and entering commands.

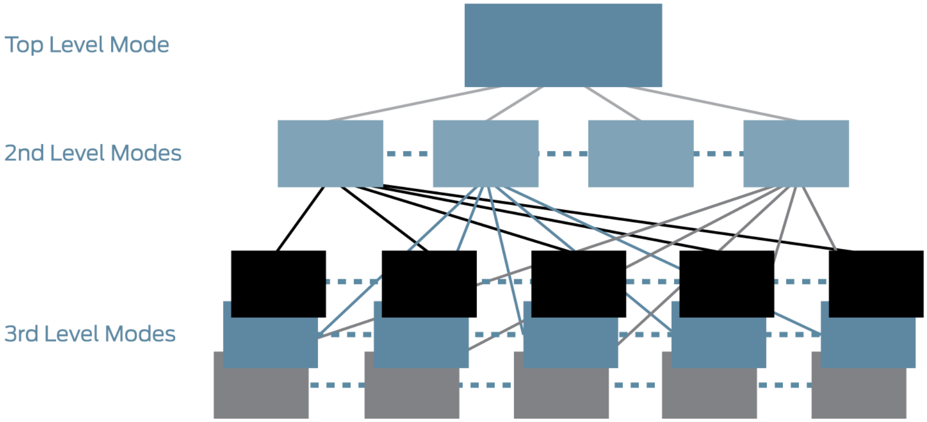


Figure 1.1 Hierarchical Structure of the Junos CLI Modes

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.

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, and more. 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.

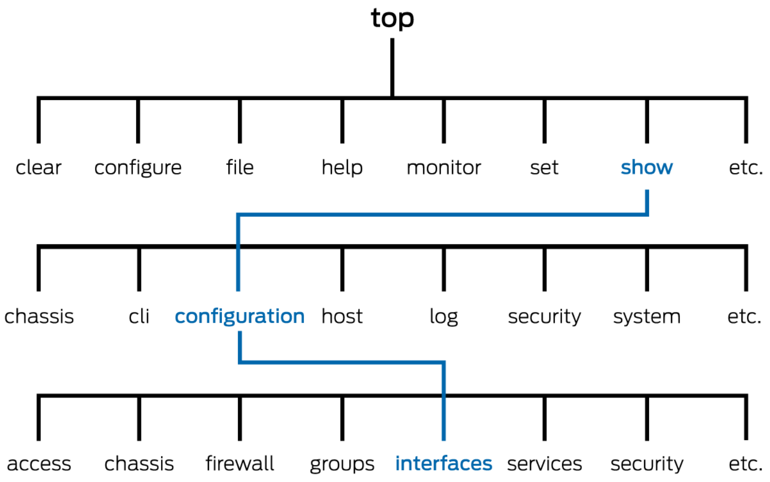


Figure 1.2 Top of the Operational Mode Tree

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 that logically groups related configuration statements. This structure eases configuration setup, review, and modification by allowing you to more readily find and view related statements. Later, in Chapter 4, you'll see how to execute operational mode commands from configuration mode with the `run` command. Figure 1.3 illustrates a portion of the configuration tree, with nodes such as system and interfaces at the second level of the hierarchy.

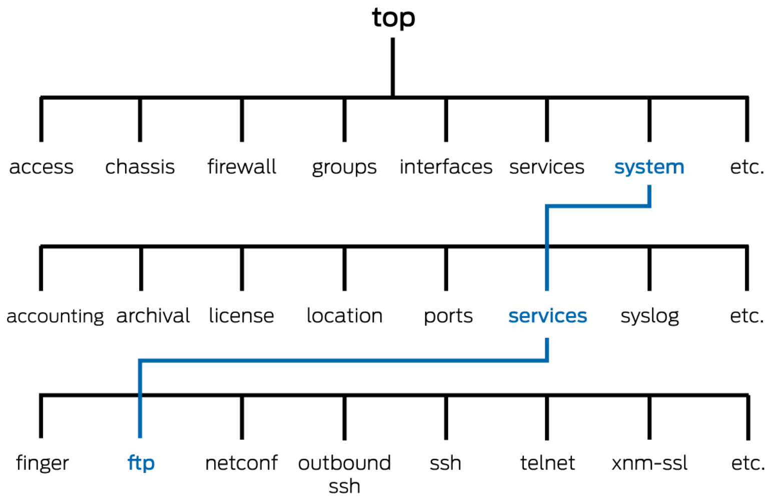


Figure 1.3 Top of the Configuration Mode Tree

The configuration statement hierarchy in the example below includes two types of statements:

- *Container statements*: contain other statements; that is, they have subordinate configuration levels. They are also called *stanzas*.
- *Leaf statements*: do not contain other statements; they are at the end of a particular hierarchical path. Leaf statements end with a semicolon (;).

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 OS CLI configuration listing:

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

- The `[edit]` banner indicates the starting hierarchy level of the listing.
- The CLI shows the hierarchy of the configuration by indenting each subordinate level.
- The CLI indicates container statements with open and closed curly braces (`{ }`). In the above example, `system` and `services` are cascading container statements.
- 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 is simply 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 in 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 example, 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 the Junos OS.

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 and practice as you go, or wait for this book's *Try It Yourself* segments.

To access the CLI, you must connect to a device and then log in. If you need help connecting to your device and logging in, see the *Quick Start* document that came with your product, or go to the URL listed in the new *MORE?* sidebar. Before logging in, you need to understand how your network is set up or have physical access to a device.

The instructions in this *Day One* book also assume that the device's management console has already been configured, and you can log in to the device using a pre-designated username 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 device at http://www.juniper.net/documentation.

NOTE If you're interested in creating new users and login accounts, you'll get to those in Chapter 7.

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 IP address of the device 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.

In many cases, telnet takes users to a shell with an % prompt. To enter the CLI, type the CLI command:

```
Last login: wed Sep 30 11:26:19 from ttsv-shell.example.com
% cli
{master:member1-re0}
user@juniper-router>
```

Switching Between Operational and Configuration Modes

As you monitor and configure a device, you will need to switch between operational mode and configuration mode. When you change to configuration mode, the command prompt also changes. The operational mode prompt is the *greater than* bracket (>). The configuration mode prompt is a *hashtag* (#).

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

```
user@juniper-router> configure
Entering configuration mode
```

You can also issue the `edit` command to enter configuration mode:

```
user@juniper-router> edit
```

SHORTCUT 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. You need to press the tab key or spacebar to use the autofill.

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

```
user@juniper-router# 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.

Try It Yourself: Moving From Configuration to Command Mode

Okay, try moving back and forth from configuration mode to command mode and back a few times using the preceding shortcut techniques.

Using Keystroke Shortcuts

The Junos OS 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.

Table 2.1 Time-saving Junos OS CLI Keyboard Shortcuts

Shortcut	Keyboard Combination
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

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]
user@juniper-router> 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
```

SHORTCUT Common abbreviations from other operating systems, such as `sh int`, are available in the Junos OS. For example:

```
user@juniper-router> sh<space>ow int<enter>
```

Try It Yourself: Using the Spacebar and Tab Key

Try entering the following operational mode 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 OS 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 Junos OS devices for years. For example, you can type `help syslog` to get help on system logs or `help tip` to get tips.

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 their usage descriptions. Typing a partial command and the question mark, `?`, provides a list of all the valid ways to complete that command. Using `?` in either of these ways is known as *context-sensitive help* in Junos OS lingo:

```
[edit system]
user@juniper-router# 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 file name as an argument, the question mark lists the files in the working directory:

```
user@juniper-router> 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:

```
user@juniper-router> 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 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 the help strings that are displayed for these:

```
[edit]
user@juniper-router# 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:

```
user@juniper-router> 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:

```
user@juniper-router> 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.
    <snip>
```

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, the Junos OS 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 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]
user@juniper-router# 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:

```
user@juniper-router> 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.

To 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>`:

```
user@juniper-router> 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.

To 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:

```
user@juniper-router> 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

```
user@juniper-router> 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 device 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
- The most common way to constrain command output is to use `pipe` | to constrain the output
- | `match`: specify exactly what information you want to display

```
user@juniper-router > 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:

```
user@juniper-router> 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:

```
user@juniper-router> 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

The Junos OS 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:

```
user@juniper-router> 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:

```
user@juniper-router> 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:

```
user@juniper-router> 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 the Shell

The kernel of the Junos OS inherits many capabilities from its UNIX roots, including the keyboard shortcuts, pipes, and expression matching discussed previously in this chapter. Another inherited functionality is the option to enter *different shells*.

When any non-root user logs in to a device running the Junos OS, the system places them in the CLI operational mode. The CLI provides access to all system management functions needed to run your system. Other shells are available to navigate the file system or for advanced recovery procedures executed by the root user, but only with the assistance of the Juniper Technical Assistance Center (JTAC).

ALERT! Use the CLI for operating the device (versus the shell) as anything outside of the CLI bypasses normal system management.

Logging In to the CLI From the Shell

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

```
% cli
```

The CLI always opens in operational mode.

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

Logging Out

You log out with the `exit` command. When you are completely logged out of the device, you receive the message: "Connection closed by foreign host."

```
user@juniper-router> 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]
user@juniper-router# exit configuration-mode
Exiting configuration mode

user@juniper-router> exit
logout
Connection closed by foreign host.
$
```

ALERT! When you exit from the standard configuration mode, all the uncommitted changes you have made during your session remain in candidate storage, unless you explicitly delete them or issue a `rollback 0` command (see Chapter 4) to reload the active configuration as the candidate. Users will get warning messages when logging in and out:

```
user@juniper-router> configure
Entering configuration mode
The configuration has been changed but not committed

user@juniper-router# exit
The configuration has been changed but not committed
Exit with uncommitted changes? [yes,no]
```

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. For more about device security, or *hardening*, see *This Week: Hardening Junos Devices, 2nd Edition*, at http://www.juniper.net/dayone.

Chapter 3

Understanding Operational Mode

Junos OS CLI 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

The Junos OS 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. All Junos OS devices come with more than sufficient processing power to collect and store critical operational data, including system logging and traceoptions that can help you to understand how the box operates in normal conditions as well as where, when, and why changes occur.

MORE? Find out more about configuring basic monitoring functions for your Junos OS device in the books of the *Day One* library. Download new titles as they become available at http://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:

```
user@juniper-router> ?
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
telnet     Telnet to another host
test       Perform diagnostic debugging
traceroute Trace route to remote host
```

Showing Device Status

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

```
user@juniper-router> show ?
Possible completions:
```

Table 3.1 lists all the possible completions of the `show ?` prompt built into the Junos OS. You may or may not know what each protocol or action is at this time. Feel free to explore in the lab using the `?` prompt.

Table 3.1 Show Commands

`accounting`	Show accounting profiles and records
`amt`	Show AMT Protocol information
`ancp`	Show ANCP information

`app-engine`	Show App-engine information
`aps`	Show Automatic Protection Switching information
`arp`	Show system Address Resolution Protocol table entries
`as-path`	Show table of known autonomous system paths
`backup-selection`	Show backup selection policies information
`bfd`	Show Bidirectional Forwarding Detection information
`bgp`	Show Border Gateway Protocol information
`bridge`	Show bridging 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
`database-replication`	Show database replication information
`ddos-protection`	Show DDOS information
`dhcp`	Show Dynamic Host Configuration Protocol information
`dhcpv6`	Show Dynamic Host Configuration Protocol v6 information
`diameter`	Show diameter information
`dot1x`	Show 802.1X information
`dvmrp`	Show Distance Vector Multicast Routing Protocol information
`dynamic-profile`	Show dynamic profile information
`dynamic-tunnels`	Show dynamic tunnel information
`esis`	Show End System-to-Intermediate System information
`event-options`	Show event-options information
`evpn`	Show EVPN information
`extension-provider`	Show extension provider parameters
`fabric`	Show RPDF Internal data structures
`firewall`	Show firewall information
`forwarding-options`	Show forwarding-options information

helper	Show port-forwarding helper information
hfrr	Show information related to Host (Direct route) Fast reroute
host	Show hostname information from domain name server
iccp	Show Inter-Chassis Control Protocol information
igmp	Show Internet Group Management Protocol information
ike	Show Internet Key Exchange information
ilmi	Show interim local management interface information
ingress-replication	Show Ingress-Replication tunnel information
interfaces	Show interface information
ipsec	Show IP Security information
ipv6	Show IP version 6 information
isis	Show Intermediate System-to-Intermediate System information
jdaf	Show JDAF information
l2-learning	Show l2 learning information
l2circuit	Show Layer 2 circuit information
l2cpd	Show l2cpd information
l2vpn	Show Layer 2 VPN information
lacp	Show Link Aggregation Control Protocol information
ldp	Show Label Distribution Protocol information
link-management	Show link management information
lldp	Show Link Layer Discovery Protocol information
log	Show contents of log file
mac-rewrite	Show layer 2 protocol tunneling information
mld	Show Multicast Listener Discovery information
mpls	Show MPLS information
msdp	Show Multicast Source Discovery Protocol information
multicast	Show multicast information
mvpn	Show Multicast Virtual Private Network (MVPN) information

mvrp	Show MVRP information
network-access	Show network-access related information
nonstop-routing	Show nonstop routing information
ntp	Show Network Time Protocol information
oam	Show Operation, Administration, and Maintenance 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 General Multicast information
pim	Show Protocol Independent Multicast information
poe	Show Power over Ethernet information
policer	Show interface policer counters and information
policy	Show policy information
ppp	Show PPP process information
pppoe	Show PPP over Ethernet information
protection-group	Show protection group information
ptp	Show Precision Time Protocol (IEEE 1588) 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	Show security information
services	Show services information
snmp	Show Simple Network Management Protocol information
spanning-tree	Show Spanning Tree Protocol information
static-subscribers	Show static-subscribers information
subscribers	Show subscriber information

`synchronous-ethernet`	Show Synchronous Ethernet related information
`system`	Show system information
`task`	Show routing protocol per-task information
`ted`	Show Traffic Engineering Database information
`unified-edge`	Show Unified-edge commands
`validation`	Show route validation information
`version`	Show software process revision levels
`virtual-chassis`	Show virtual chassis information
`vpls`	Show VPLS information
`vrrp`	Show Virtual Router Redundancy Protocol information

TIP For the reader with experience using Cisco IOS software, one of the basic differences between Cisco IOS and Junos OS is that Junos OS does not use the keyword `IP`, so many of the show commands you already know will work if you drop this part of the command. For example, the IOS command `show ip route` simply becomes `show route` in the Junos OS.

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:

```
user@juniper-router> show interfaces fe-1/1/1 ?
Possible completions:
```

Table 3.2 lists the possible completions built into the Junos OS.

Table 3.2 Common Show Command Arguments

< 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

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

```
user@juniper-router> 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
```

```
 user@juniper-router> 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.

Try It Yourself: get terse

If you are following this book in your lab, try the various command arguments `terse`, `detail`, and `extensive`.

Managing Basic Operations

Junos 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:

```
user@juniper-router> 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:

```
user@juniper-router> 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.

You are not limited to copying files on the same device. You can use the `same` command to copy files to and from a file server. And here is how you would move the configuration file from the server back to the device's home directory:

```
user@juniper-router> file copy username@server-host-name:/config/juniper.config.gz /
var/home/user/juniper.config.gz
```

The file list Command

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

```
user@juniper-router> file list

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

You can specify a directory with the `file list` command if the file is not in your home directory (and you have access to the folder):

```
user@juniper-router> file list /var/tmp

/var/tmp/:
.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 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 processes from the operational mode. 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.

```
user@juniper-router# deactivate protocols ospf
```

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:

```
user@juniper-router> 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? Junos technical documentation provides details about upgrading the software version of your device. You can download the current package from the software download page at http://www.juniper.net/support. Note that downloading new software requires a current service contract and login account.

MORE? To learn more about operational mode commands, see the Juniper TechLibrary at: http://www.juniper.net/techpubs/en_US/junos12.1/information-products/pathway-pages/swcmdref/swcmdref.html.

Chapter 4

Discovering Configuration Mode

In configuration mode, as the name implies, you define the configuration on 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 OS is thoughtfully designed with configuration set up as a multistep process. For example, safeguards allow you to create and check a new configuration before it goes live. The Junos OS captures all changes in a candidate configuration; the candidate becomes the active running configuration only when you are ready and only when you enter a `commit` command.

This approach significantly 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 feedback from early adopters, 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.

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

Figure 4.1 illustrates the three basic steps to configure a device running the Junos OS.

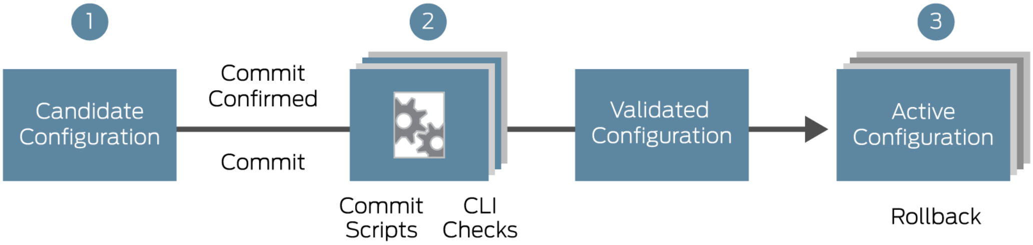


Figure 4.1 Junos CLI Configuration Process Overview

Here is an explanation of the steps identified in Figure 4.1:

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, J-Web interface, or by automated means. Junos also includes commands to review your candidate changes, 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 validation 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.

NOTE The Junos OS 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. The Junos OS 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 all users can see them.
- *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 containing only the changes by the private user.

The configure Command

Enter the standard configuration mode and issue the `configure` command:

```
user@juniper-router> configure
Entering configuration mode
```

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 non-committed changes to the configuration once you exit the session:

```
user@juniper-router> 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:

```
user@juniper-router> configure private
warning:uncommitted changes will be discarded on exit
Entering configuration mode
```

When a private user commits changes, Junos 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 non-conflicting 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, if an administrator accidentally typed the `delete interfaces` command and recognized the mistake but instead of removing the statement, simply exited configuration mode, later, when another user logged in and committed the configuration, all the device's interfaces would be deleted! Fortunately, the Junos OS makes it possible to roll back to a previous configuration.

Understanding Configuration Mode Basics

Configuration mode also offers several options to view and navigate the candidate configuration as you proof and verify changes before any kind of commit.

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]
user@juniper-router# show
version “14.2R1.3”;
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 that current configuration hierarchy level and below:

```
[edit interfaces ge-5/0/0]
user@juniper-router# show
gigether-options {
  flow-control;
  auto-negotiation;
}
unit 0 {
  family inet {
    address 10.2.3.4/28;
  }
}
```

NOTE You may have noticed that the configuration mode uses the show command in a different way from operational mode. The commands of each mode are independent of each other, and so the 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]
user@juniper-router# set system services finger
user@juniper-router# set system services ftp
user@juniper-router# set system services ssh
user@juniper-router# set system services telnet
```

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

```
[edit system services]
user@juniper-router# set finger
user@juniper-router# set ftp
user@juniper-router# set ssh
user@juniper-router# 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]
user@juniper-router# edit system services

[edit system services]
user@juniper-router#
```

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]
user@juniper-router# edit system syslog host log

[edit system syslog host log]
user@juniper-router#
```

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

```
[edit]
user@juniper-router# edit system

[edit system]
user@juniper-router# edit syslog

[edit system syslog]
user@juniper-router# edit host log

[edit system syslog host log]
user@juniper-router#
```

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 within 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]
user@juniper-router# up

[edit interfaces fe-1/3/1 unit 0 family inet]
user@juniper-router#
```

In this 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 highest hierarchy location from which you previously entered an 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]
user@juniper-router# top edit system login

[edit system login]
user@juniper-router#
```

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

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

MORE? To learn more about using set commands to configure your device, see any of the *Day One* books in the *Junos Fundamentals Series*. They all contain many examples. Download a few books and see if you can read and understand the configuration examples at http://www.juniper.net/dayone. *Day One* eBoooks are also available at many mobile device bookstores.

Editing the Configuration

You can, of course, 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 issue the following set command:

```
[edit]
user@juniper-router# set system services ftp
```

The following lines are added to the configuration file:

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

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

```
[edit]
ser@juniper-router# 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 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]
user@juniper-router# 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 an 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]
user@juniper-router# set interfaces ge-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]
user@juniper-router# show

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

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

```
[edit]
user@juniper-router# delete interfaces ge-1/0/0
```

4. Use the `show` command to verify that the ge 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 by allowing the device to search through the entire candidate configuration looking for a string and delete every line that contains that string. To learn more about wildcards, see the *Junos CLI User Guide* at: https://www.juniper.net/techpubs/en_US/junos15.1/information-products/pathway-pages/junos-cli/junos-cli.html.

The annotate Command

The Junos OS 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. This example from the top of the configuration mode posts the comment at the `[edit system]` level of the configuration hierarchy:

```
[edit]
user@juniper-router# annotate system  “this device is for training new users”
```

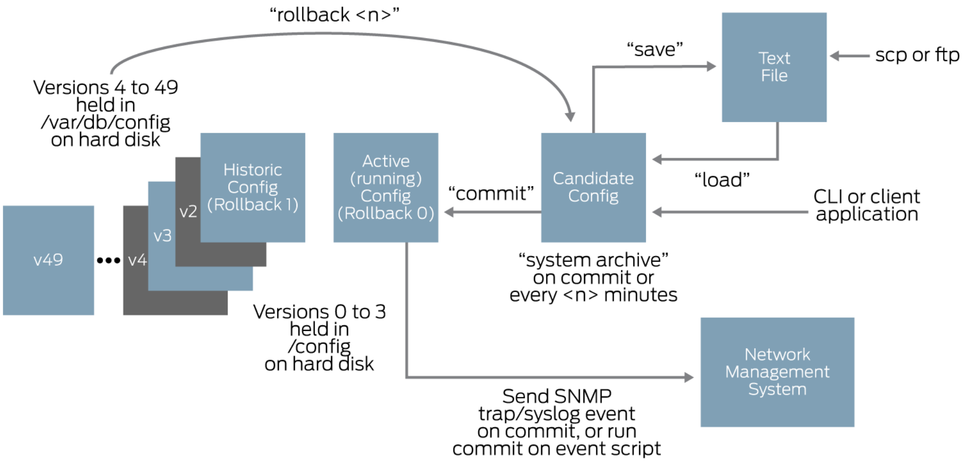
When you add comments in configuration mode, they are associated with a statement at the indicated level. Each statement can have one, single-line, comment associated with it. To delete a comment, use the `annotate` command with an *empty* string:

```
[edit]
user@juniper-router# annotate system “ ”
```

Committing the Candidate Configuration

The Junos OS CLI also provides multiple features that help users 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 schematic of the file management of the device configuration.



“rollback <n>”
“save”
Text File
scp or ftp
Versions 4 to 49 held in /var/db/config on hard disk
Historic Config (Rollback 1)
Active (running) Config (Rollback 0)
“commit”
Candidate Config
“load”
CLI or client application
v49
v4
v3
v2
“system archive” on commit or every <n> minutes
Versions 0 to 3 held in /config on hard disk
Network Management System
Send SNMP trap/syslog event on commit, or run commit on event script

Figure 4.2 Junos OS Configuration File Management

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 that pass the validation checks):

- Copies the candidate configuration to the active configuration. At this point, the active and the candidate configurations are identical.
- Decrements all rollback configuration files by one, and saves the active configuration as rollback 0.

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 on-box access to a total 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]
user@juniper-router# set services telnet

[edit system]
user@juniper-router# delete services web-management

[edit system]
user@juniper-router# delete services ssh
```

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

```
[edit system services]
user@juniper-router# 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.

SHORTCUT

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

```
[edit]
user@juniper-router# 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]
user@juniper-router# 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 proof 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 a *proposed* configuration, and your device does not use any of it 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]
user@juniper-router# commit
commit complete
```

Before actually activating the candidate configuration, Junos 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]
user@juniper-router# 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]
user@juniper-router# 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.).

The commit confirmed Command

Ever make the mistake of adding security only to discover the new firewall locked you out of the very interface that you were using to access the device? Ever accidentally isolate a remote box and then have to jump in the car and drive for 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 ten 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 any time you want a safety net against potential configuration problems:

```
[edit]
user@juniper-router# commit confirmed
commit confirmed will be automatically rolled back in 10 minutes unless confirmed
commit complete
```

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

```
[edit]
user@juniper-router# 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 previously active configuration at the end of the ten minutes. In this way, if you have accidently isolated a 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@juniper-router
         (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 confirmed` command again.

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

```
[edit]
user@juniper-router# commit confirmed 2
commit confirmed will be automatically rolled back in 2 minutes unless confirmed
```

BEST PRACTICE When you are configuring remote devices, *always* use the `commit confirmed` command to activate your candidate configuration. Even the most experienced Junos OS 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.

Rolling Back the Configuration

Whenever you commit the candidate configuration as the new active configuration, Junos 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]
user@juniper-router# rollback 1
load complete
```

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

```
[edit]
user@juniper-router# show
<snip>

[edit]
user@juniper-router# 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 in order 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 it with the show | compare command:

```
[edit interfaces]
user@juniper-router# 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]
user@juniper-router# rollback ?
Possible completions:
<[Enter]>              Execute this command
0           2009-01-31 04:34:56 UTC by walter via cli
1           2009-01-31 04:30:03 UTC by walter via cli
2           2009-01-30 06:23:44 UTC by walter via cli
<snip>
48          2008-11-03 08:00:03 UTC by walter via cli
49          2008-11-03 07:45:21 UTC by walter 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 OS CLI provides options for making system changes in advance, a neat way to prepare yourself for the next cutover or maintenance window. These options will remain inactive parts of your configuration until needed, or they could become scheduled elements with set activation times. Good stuff!

Preconfiguration of New Hardware

Junos OS can prepare you 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 the interface fe-3/0/0, for example, which could be installed tomorrow:

```
[edit]
user@juniper-router# edit interfaces fe-3/0/0 unit 0

[edit interfaces fe-3/0/0 unit 0]
user@juniper-router# set family inet address 192.168.1.254/24

[edit interfaces fe-3/0/0 unit 0]
user@juniper-router#
commit complete
```

The commit at Command

You might 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]
user@juniper-router# commit at 02:00:00
commit check succeeds
commit will be executed at 2009-02-02 02:00:00 UTC
Exiting configuration mode
user@juniper-router>
```

To display any pending commit operations (and the commit history), enter the `show system commit` command. If you see something pending that you don't like, you can cancel the pending commit operation with the `clear system commit` command:

```
user@juniper-router> 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 if they were not even defined. In this example, a new BGP neighbor at 192.168.1.1 is configured but left deactivated until the session is ready to be introduced:

```
[edit]
user@juniper-router# edit protocols bgp group internal

[edit protocols bgp group internal]
user@juniper-router# set neighbor 192.168.1.1

[edit protocols bgp group internal]
user@juniper-router# deactivate neighbor 192.168.1.1

[edit protocols bgp group internal]
user@juniper-router# 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]
user@juniper-router# 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]
user@juniper-router# activate neighbor 192.168.1.1

[edit protocols bgp group internal]
user@juniper-router# commit
commit complete
```

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

```
[edit protocols bgp group internal]
user@juniper-router# up

[edit protocols bgp]
user@juniper-router# deactivate group internal
```

You must commit the configuration to reflect these changes.

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 can then modify that second element with any needed changes.

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

```
[edit interfaces]
user@juniper-router# show
xe-0/0/2 {
  unit 0 {
    family inet {
      address 10.0.24.13/30;
    }
    family mpls;
    }
}
```

You can use the copy command to set up a new interface, xe-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 xe-0/0/1, for example, you might change its address:

```
[edit interfaces]
user@juniper-router# copy xe-0/0/2 to xe-0/0/1

[edit interfaces]
user@juniper-router# delete xe-0/0/1 unit 0 family inet address 10.0.24.13/30
```

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

```
[edit interfaces]
user@juniper-router# set xe-0/0/1 unit 0 family inet address 10.0.36.2/30
```

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

```
[edit interfaces]
user@juniper-router# show
xe-0/0/1 {
  unit 0 {
    family inet {
      address 10.0.36.2/30;
     }
    family mpls;
   }
}
xe-0/0/2 {
  unit 0 {
    family inet {
      address 10.0.24.13/30;
     }
    family mpls;
   }
}
```

ALERT! Before committing 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]
user@juniper-router# 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]
user@juniper-router# 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]
user@juniper-router# 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 OS, 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]
user@juniper-router# 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]
user@juniper-router# 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 Find out about configuration templates and other shortcuts in Chapter 9. To learn more about the CLI, see the Juniper TechLibrary at: https://www.juniper.net/techpubs/en_US/junos15.1/information-products/pathway-pages/junos-cli/junos-cli.html.

Chapter 5

Creating a Checklist

If you read the first four chapters of this *Day One* book, you should have a good idea of how to use the Junos OS CLI, and you should have acquired sufficient basic navigation skills and configuration basics to set up a new router, a switch, or a security platform. *It's Day One and you have a job to do!* So let's use your newfound knowledge and go through the steps entailed in configuring device basics using the Junos OS CLI.

Checklist Components

Of course the steps to device configuration need to follow common practices. And the first best practice has nothing to do with the CLI. It's getting a list of the information you need together *before* you start to configure device basics. There's a table at the end of this chapter that you can use to record this information, one way or another, so you have an available record of vital information, not a bunch of numbers in your head. Also, even the old-timers sometimes use a pen and paper because it's easier when you're crawling among the racks.

By creating a checklist, and filling it out, you will breeze through the basic device setup discussed throughout the remaining chapters of this book.

Hostname

Most devices in your network infrastructure – whether they are routers, switches, servers, or firewalls – are known by a specific name rather than by the device's IP address. A name is simply easier to remember than a long string of numbers. The device's name is also known as the *hostname*.

Often, administrators choose a hostname that reflects the device's use in the network, for example: *uk-london-R1*. The hostname should be unique to the device. It is usually added to the DNS server so administrators can connect more easily to the device using an easy-to-remember hostname.

Loopback Interface

Most of the addresses you configure on your device are physical interfaces, however, the *loopback interface* is a virtual interface – an interface not associated with any hardware or network. While physical interfaces might be removed or their addresses changed, the loopback address never changes. The loopback address has many different uses in the operation and management of the network.

Management Interface

A *management interface* lets authorized users and management systems connect to the device over the network. Your device may have a dedicated management port on the front panel, or you can configure a management interface on one of the network interfaces. This interface can be dedicated to management or shared with other traffic.

You can configure your device to use *in-band management, out-of-band management*, or both. Out-of-band management uses a dedicated management port to limit access to the device by making it only possible over a dedicated and separate management network that does not carry user traffic. In-band management allows users to access the device over links also used for regular traffic and thus requires very strict security to prevent unauthorized intrusions. This section uses out-of-band management, which is considered more secure.

Before users can access the management interface, you must configure it. Information required to set up the management interface includes its IP address and prefix. In many types of devices that run the Junos OS (or recommended configurations) it is not possible to route traffic between the management interface and the other ports. Therefore, you should select an IP address in a separate (logical) network, with a separate prefix (netmask).

Backup Router

If the Junos OS is running on a network device that performs Layer 3 forwarding (such as a router) you may want to specify a backup router. A backup router can be used during the initial boot process of the Junos OS, before any routing protocols have converged. It allows the device to establish a Layer 3 connection quickly, thus keeping unavailability to a minimum. In selecting a backup router, it is common practice to choose the default gateway of your management network that is directly connected to your device.

Domain Name System (DNS)

It is easier for most people to remember names than numbers, especially if those numbers are IPv4 addresses. Because of this, DNS servers are used to map device hostnames to IP addresses and vice versa.

With DNS you can use names to designate key external systems such as file and log servers that your device may need to contact. The DNS server maintains a centralized repository for device hostnames on the network, ensuring each device hostname is unique. This centralized repository makes it easier to query and to administer translations between the network IP addresses and hostnames. You can configure your device to query one or more DNS servers via their IP addresses.

Time Servers

The IETF specified the Network Time Protocol (NTP) to synchronize the clocks of computer systems connected to each other over a network. Most large networks have an NTP server that ensures that time on all devices is synchronized, regardless of the device location. If you have to use NTP server(s) on your network, ensure you know its (their) address(es).

User Names and Passwords

The root account of the Junos OS provides full administrative access to your device with complete control over its configuration and operation. The root account is often referred to as the *superuser*.

In new devices, the root account has no password. You must add a password to the root account before you can commit any configuration. The stronger you make the password, the harder it is for others to discover it and use it to break into the account.

The Junos OS helps to enforce the use of strong passwords. For example, valid passwords must:

- Be a minimum length of six characters.
- Contain at least one change of case or character class.
- Be at least six characters long and use at least three of the five defined character classes (uppercase letters, lowercase letters, numbers, punctuation marks, and other special characters).

Control characters are not recommended for passwords.

BEST PRACTICE Increase the length of the password and the minimum number of case, digit, and punctuation changes to set up safer passwords. An example of a good password would be: *t3aMX*u7rS*.

In addition to the root user, it is highly recommended that you create *at least* one other local user. This user can log in when you need to perform administration or maintenance tasks on the device.

In assigning usernames, do not include spaces, control characters, colons, or commas. A username can be a maximum length of up to 64 characters. User passwords also require a change of case, digits, or punctuation.

Remote Authentication Servers

You probably already use a remote authentication server (or servers) in your network. It's a recommended best practice, because it allows you to centrally create a consistent set of user accounts for all devices in your network.

Using a central server has multiple advantages over the alternative of creating local users on each and every device – a time-consuming and error-prone task. A central authentication system also simplifies the use of one-time password systems such as SecureID, which offer protection against password sniffing and password replay attacks, in which someone uses a captured password to pose as a system administrator.

There are two basic methods of remote authentication in use by most enterprises today: RADIUS (Remote Authentication Dial-in User Service) and TACACS+ (extended Terminal Access Controller Access Control System Plus). The Junos OS can be configured to query multiple remote authentication servers of both types.

MORE? If you want more information about RADIUS or TACACS+ technologies, see *Configuring System Features* in the Juniper TechLibrary at https://www.juniper.net/documentation/en_US/junos/information-products/pathway-pages/system-basics/index.html.

Network Interfaces

If your device is performing Layer 3 forwarding (for example, IP routing), it needs at least one IP address assigned to an interface. If you have more than one network interface, you need at least one IP address for each. This book will help you to configure a Gigabit Ethernet interface so that you can get your device "up and running" on your network.

MORE? The Junos OS documentation contains information on how to configure other types of interfaces. See the Juniper TechLibrary for more information at: https://www.juniper.net/techpubs/en_US/junos15.1/information-products/pathway-pages/config-guide-network-interfaces/network-interfaces.html.

System Logging

You may also want to configure remote logging on your device. Junos OS uses a syslog mechanism similar to many UNIX devices to forward log messages to a specified log host address. This allows each of your devices to forward their messages to one central host, making it easier to monitor the network as a whole. Syslog is a very flexible and rich way of logging messages and is used by many device vendors to supplement the information provided by SNMP traps.

That's the End of the Gathering

That's it. That's the information you need to begin your configuring. Use Table 5.1 on the next page to record the information you've gathered. Now you have a reference to complete the commands and steps described in the remaining chapters of this book.

Keep it handy!

Table 5.1 Configuration Information Worksheet (printable)

Category	Device
Hostname	
Management Port IP Address	
Management Port Network Prefix	
Loopback Interface IP Address	
Backup Router IP Address	
Domain Name Server IP Addresses	
Network Time Protocol Server IP Address	
Initial Root Password	(If you write down a password here, make sure that this document is secure.)
Local Username	
Initial Local Password	
Method of Remote Authentication You Use	
IP Address(es) of Your Remote Authentication Server(s)	
Server Authentication Password	(The device may need a secret key for encryption to access the authentication server.)
IP Address of Interface(s)	
IP Address of a Network Management System	
SNMP Community	
IP Address of a Log Host	

Chapter 6

Configuring System Basics

Okay, it's time to actually set up your device with the basic settings, including the base system, user accounts, remote access, and interfaces.

As you follow along, customize the command entries using the specific information gathered in your *Configuration Information Worksheet* (Table 5.1). This will ensure connectivity in your own lab.

Configuring Base System Settings

The first steps in configuring your device are the base system settings:

- root (administrator) password
- hostname
- management interface
- loopback interface
- backup router

NOTE This book follows the convention of not always showing the command prompt in displaying configuration mode command examples.

TIP If you explore the Junos OS [edit system] hierarchy, here's a quick reminder of the basic settings that you can configure:

```
[edit system]
root@juniper1# set system ?
Possible completions:

+ authentication-order Order in which authentication methods are invoked
> backup-router        IPv4 router to use while booting
  domain-name          Domain name for this router
  host-name            Hostname for this router
> location             Location of the system, in various forms
> login                Names, login classes, and passwords for users
> name-server          DNS name servers
> ntp                  Network Time Protocol services
> radius-options       RADIUS options
> radius-server        RADIUS server configuration
> root-authentication  Authentication information for the root login
> syslog               System logging facility
time-zone              Time zone name
<snip>
```

Root Authentication Password

The root account or user is a predefined user name in the Junos OS. The root user is by default the administrator or superuser, who has absolute permission to both configure and install software on the device.

The Junos OS requires configuration of the root password before it accepts a commit. On a new device, the root password must always be a part of the configuration submitted with your initial commit. Use the following command to set up a plain text password for the root user:

```
root@juniper1# set system root-authentication plain-text-password
New password: ######
Retype new password:  ######
```

As you enter the password in plain text, the software encrypts it immediately – you don't have to configure the Junos OS to encrypt the password as in some other systems. Plain text passwords are therefore hidden and marked as `## SECRET-DATA` in the CLI configurations.

BEST PRACTICE Strengthen security by only allowing root access from the console port:

```
root@juniper1# set system services ssh root-login deny
```

Hostname

The hostname of the device provides its identification for many purposes and the Junos OS uses the configured hostname as part of the command prompt, to prepend log files and other accounting information, as well as in other places where knowing the device identity is useful. Let's use `juniper1`, but you can choose anything more aptly descriptive:

```
root@juniper1# set system host-name juniper1
```

Loopback Interface

The loopback interface supports many different network and operational functions and is an "always up" interface. For example, the loopback interface assures that the device is reachable, even if some of the physical interfaces are down, removed, or an IP address has changed. In most cases, you always define a loopback interface.

The Junos OS follows the IP convention of using lo0 as the identifier name of the loopback interface. Refer to your worksheet (Table 5.1) as a reminder of what you have chosen as the IP address of your loopback interface:

```
root@juniper1# set interfaces lo0 unit 0 family inet address 192.26.0.110/32
```

NOTE See the section *Introducing Interfaces* in this chapter for more information about the `set interfaces` command format.

ALERT! The Junos OS requires that the loopback interface always be configured with a /32 network mask (avoiding any unnecessary allocation of address space).

You can configure as many addresses as you need on the lo0 interface, so it's good practice to make one address preferred:

```
root@juniper1# set interfaces lo0 unit 0 family inet address 192.26.0.110/32 preferred
```

Only `unit 0` (unit is a reference to a logical channel on Junos interfaces) is permitted as the master loopback interface. If you want to add more IP addresses to this, you simply configure them in the normal way under unit 0, without the `preferred` option:

```
root@juniper1# set interfaces lo0 unit 0 family inet address 192.168.1.1/32
root@juniper1# set interfaces lo0 unit 0 family inet address 192.168.2.1/32
```

BEST PRACTICE On the lo0.0 interface, it is useful to have the IP address 127.0.0.1 configured, as certain processes such as NTP and MPLS ping use this default host address:

```
root@juniper1# set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

The 127.0.0.1/32 address is a *Martian IP address* (an address invalid for routing), so it is never advertised by the Juniper device.

NOTE Depending on your network configuration, you may also need an ISO address for the IS-IS routing protocol:

```
root@juniper1# set interfaces lo0 unit 0 family iso address 49.0026.0000.0000.0110.00
```

Management Interface

The management interface supports access to your device for authorized users as well as management systems. Users can then connect to the management interface over the network using standard utilities such as SSH and Telnet (See Chapter 7).

Many types of Junos OS physical platforms include a dedicated management port on the front panel. For those that do not, you can configure one of the Ethernet ports to act as the management interface.

A network interface can be configured as being dedicated to out-of-band management or as being shared by both management and network traffic.

NOTE Even though your device has a dedicated management port, you may prefer to configure a network interface to carry management traffic. For example, your organization may use this approach when cost does not justify a separate management infrastructure.

Dedicated Management Port

The dedicated management port supports out-of-band management access with complete physical separation from network traffic within your device. This approach limits access to your device, and thereby the potential for problems. Further, because it only carries management traffic, the management port is fully available to you for analyzing and reacting to problems if your device happens to be under attack.

How to set up the dedicated management port:

Configuration of a dedicated management port simply requires assignment of the IP address that you want to use as the management interface. The interface name that you use in the Junos OS CLI command depends upon the type of device that you are setting up.

The following example shows the command format to set up the dedicated management port for an EX Series switch. The EX Series Ethernet Switches use the interface name *me0* as the name of the management port:

```
root@juniper1# set interfaces me0 unit 0 family inet address 172.26.27.44/24
```

On other devices running the Junos OS, the dedicated management port is named *fxp0*. Table 6.1 outlines the assigned names of the dedicated management port in various Juniper platforms. If you are using one of these other platforms, substitute the interface name `me0` with `fxp0` in the above command statement.

Table 6.1 Names of Dedicated Management Ports

Platform	Dedicated Management Port
EX Series Ethernet Switches	me0 (see additional note below)
MX Series, T Series Routers	fxp0
PTX Series Routers	em0
QFX Series Switches	c0, c1, etc.
SRX5xxx and SRX3xxx Services Gateways	fxp0

NOTE The EX Series also includes a commonly used virtual management Ethernet interface known as *VME* that is used for managing devices grouped together in a virtual chassis. For more information see *Day One: Configuring EX Series Ethernet Switches, 3rd Edition*, at http://www.juniper.net/us/en/training/jnbooks/day-one/fabric-switching-tech-series/config-ex-series/.

Management Over a Network Interface

If your device uses a network interface for carrying management traffic, you need to similarly configure it with the IP address that you want to use as the management interface.

The following sections discuss how to configure the management interface on a branch SRX Series device or J Series in flow-based mode. When configuring these platforms, you must assign the configured management interface to a zone before it can carry traffic. The zone provides virtual separation of traffic and acts as a policy enforcement point.

How to set up management on a dedicated network interface with zones:

Use the following commands to set up the management interface on a network interface that is dedicated to out-of-band management. As a reminder, these specific steps are for a SRX Series branch device or a J Series in flow-based mode.

1. Configure the interface with the IP address that you are using for management:

```
root@juniper1# set interfaces ge-0/0/0 unit 0 family inet address 172.26.27.44/24
```

2. Before the interface can carry traffic, you need to set up the zone. The functional zone management is a special predefined zone for out-of-band management in these platforms. Add the logical interface to this zone with the following command:

```
root@juniper1# set security zone functional-zone management interfaces ge-0/0/0.0
```

3. Where you have set up a functional zone, it is necessary to specify which protocols the interface responds to, for example:

```
root@juniper1# set security zone functional-zone management host-inbound-
traffic system-services ssh
```

How to set up management on a shared network interface with zones:

An alternative to using a dedicated out-of-band interface is to share a network interface between management and revenue traffic. You may prefer this approach if you are configuring a branch device with a limited number of ports, and don't want to dedicate an entire port just to management.

Use the following steps to set up the management interface on a shared network interface in a SRX Series branch device or a J Series in flow-based mode.

1. Configure the interface with the IP address that you are using for management:

```
root@juniper1# set interfaces ge-0/0/0 unit 0 family inet address 172.26.27.44/24
```

2. Before the interface can carry traffic, you need to set up the zone. The `security-zone trust` is a pre-defined security zone. Add the logical interface to this zone with the following command:

```
root@juniper1# set security zone security-zone trust interfaces ge-0/0/0.0
```

3. Specify which protocols the interface in the `trust` zone responds to, for example:

```
root@juniper1# set security zone security-zone trust host-inbound-traffic system-
services ssh
```

Backup Router

Okay, back to our basic configuration list.

You can configure the Junos OS to use a backup router during the initial boot process in Layer 3 devices. The process responsible for establishing routes (among other functions) is known as the routing protocol daemon (RPD). When the software is booting, RPD is not initially running, and therefore the device has no routes. Configuring a backup router allows the device to establish a Layer 3 connection quickly during boot time, thereby minimizing the amount of time the device is unavailable:

```
root@juniper1# set system backup-router 172.26.31.1 destination 172.16.0.0/12
```

Here, in this example, if you choose the default gateway of your management system, the management network (all of the IP range 172.16/12) is reachable via next-hop 172.26.31.1 early on in the boot process, even before other routing protocols have converged.

NOTE The Junos OS only uses the backup router during the boot sequence. If you want to configure a backup router for use after startup, you can set up a default route, but that is beyond the scope of this *Day One* book. See the Juniper TechLibrary for more specific information: http://www.juniper.net/documentation/en_US/junos15.1/topics/task/configuration/backup-router-configuring.html.

Reaching a Domain Name System (DNS) Server

The Junos OS can resolve hostnames to IP addresses if it knows the location of your DNS server(s). The approach is similar to the way a web browser resolves the name of a website to its network address.

Additionally, Junos OS lets you configure one or more domain names, which it uses to resolve hostnames that are not fully qualified (therefore, the domain name is missing). This is convenient as you can simply use a hostname in configuring and operating the operating system without the need to reference the full domain name.

SHORTCUT After adding a DNS server(s) and domain name(s) to your configuration, you can use DNS resolvable hostnames in your configuration and commands instead of IP addresses.

How to configure the DNS server:

Configure the Junos OS to use one or more DNS servers.

1. Begin by including the IP address(es) of your DNS server(s)within a name-server statement(s):

```
root@juniper1# set system name-server 172.26.27.2
root@juniper1# set system name-server 172.26.27.3
```

2. It's good practice to configure the domain name in which the device itself is located. The Junos OS then uses this configured domain name as the default domain name to append to hostnames that are not fully qualified:

```
root@juniper1# set system domain-name enterprise.com
```

3. If your device can reach several different domains, you can configure these as a list of domains to be searched. The Junos OS then uses this list to set an order in which it appends domain names when searching for the IP address of a host:

```
root@juniper1# set system domain-search [enterprise.com department.enterprise.com]
```

This command tells the Junos OS to search the enterprise.com and then the department.enterprise.com domains when attempting to resolve unqualified hosts.

4. If you have configured your DNS server with the hostname and an IP address for your device, you can issue the following commands to confirm that DNS is working and reachable.

In the first command use the IP address of your device to confirm resolution to the configured hostname:

```
root@juniper1> show host 172.26.27.44
44.27.26.172.in-addr.arpa domain name pointer juniper1.enterprise.com.
```

In the second command, use the configured hostname to confirm resolution to the IP address:

```
root@juniper1> show host juniper1
juniper1.enterprise.com has address 172.26.27.44
```

NOTE It doesn't matter which IP address you assign as the address of your device in the DNS server, as long it is an address that reaches your device. Here we have used the management interface, but you may choose the loopback interface, a network interface, or even configure multiples of the addresses on the DNS server.

Setting Up the Date and Time

The initial configuration of a device should include time settings for accurate recording of events. To set the time in your device running the Junos OS, you can either configure it manually, or your device can take a system time from an NTP server.

How to set time locally:

If you do not have access to an NTP server you can configure the Junos OS to keep its own local time using an onboard clock. You can manually configure the date and time from operational mode:

```
root@juniper1# set date 200901011200.00
```

The date format is YYYYMMDDhhmm.ss.

How to use a remote time server:

In large and small networks it's preferable to have access to an NTP server to set the exact same time across all the network devices. The common reference lets you correlate the timestamps of logs and trace files for troubleshooting purposes.

1. The easiest way to have NTP set the time is to have the Junos OS retrieve the time when it first boots up. Use the following command with the IP address of your NTP server:

```
root@juniper1# set system ntp boot-server 172.26.27.4
```

2. To keep the device synchronized with periodic updates, configure a reference NTP server (you can configure more than one). It's good practice to do this, as the Junos OS device can be up and running for a long time, and therefore the clock can drift:

```
root@juniper1# set system ntp server 172.26.27.4
```

3. Next, you may want set the local time zone to match the device's location (note that Universal Coordinated Time [UTC] is the default). This allows the Junos OS to present the time in the correct local format, accounting for things such as offset from UTC, which may change several times throughout the year:

```
root@juniper1# set system time-zone Europe/Amsterdam
```

TIP Many administrators prefer to keep all their devices configured to use the UTC time zone. This approach has the benefit of allowing you to easily compare the time stamps of logs and other events across a network of devices in many different time zones.

4. If you’ve just booted the Junos OS and need to synchronize time with a remote time source, here’s how to do so in operational mode:

```
root@juniper1# set date ntp 172.26.27.4
7 Apr 10:32:27 ntpdate[4544]: step time server 172.26.27.4 offset -0.000565 sec
```

How to verify your time settings:

After you set up the time, you can check the configuration in the following ways.

1. Check the system time at any time (pardon the pun):

```
root@juniper1# show system uptime
Current time: 2009-04-06 15:36:10 CEST
System booted: 2009-03-27 12:56:33 CET (1w3d 01:39 ago)
Protocols started: 2009-03-27 12:58:04 CET (1w3d 01:38 ago)
Last configured: 2009-04-06 15:27:02 CEST (00:09:08 ago) by username
 3:36PM  up 10 days,  1:40, 1 user, load averages: 0.00, 0.00, 0.00
```

This listing provides not only the current time, but also when the device was last booted, when the protocols were started, and when the device was last configured.

2. You can also check the NTP server status and associations of the clocking sources used by your device with the following two commands:

```
root@juniper1> show ntp associations

     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*172.26.27.4     203.26.24.6      3 u   16   64  377    0.256   -0.164   0.022

root@juniper1> show ntp status
status=0644 leap_none, sync_ntp, 4 events, event_peer/strat_chg,
version=”ntpd 4.2.0-a Wed Mar 25 07:21:19 UTC 2009 (1)”,
processor=”i386”, system=”JUNOS9.4R2.9”, leap=00, stratum=4,
precision=-19, rootdelay=502.545, rootdispersion=74.632, peer=59484,
refid=172.26.27.4,
reftime=cd847de9.ccb54775  Mon, Apr  6 2009 15:11:05.799, poll=6,
clock=cd847dfc.4a08cfa9  Mon, Apr  6 2009 15:11:24.289, state=4,
offset=-0.164, frequency=52.814, jitter=0.030, stability=0.005
```

Introducing Interfaces

The interfaces available in devices running the Junos OS include physical interfaces for moving traffic in and out of the device, as well as special interfaces such as the management and loopback interfaces discussed a few pages back.

As part of the basic setup of a device, this section discusses the format of the interface naming, introduces logical interfaces, and shows how to configure a Gigabit Ethernet interface.

MORE? For more on how to configure other types of interfaces see the Juniper TechLibrary at: https://www.juniper.net/techpubs/en_US/junos15.1/information-products/pathway-pages/config-guide-network-interfaces/network-interfaces.html.

Physical Interfaces

Your platform may include Ethernet interfaces and one or more of many WAN interfaces available on different types of devices running the Junos OS. Regardless of the type of interface, the software follows a standard format in its interface naming. The interface name is made up of two parts: the *interface type* and the *interface numbering*.

Interface Type

The Junos OS denotes the different types of interfaces with a text identifier. For example, the identifier for a Gigabit Ethernet interface is the text string *ge*.

Interface Numbering

Juniper Networks engineering assigns interface numbers corresponding to each interface location (in each hardware platform), such as *ge-0/0/1*. Generally, they use the following conventions in designating numbers to the device interfaces, sequentially assigning numbers beginning with zero (0).

- *slot:* the first number corresponds to the slot location. On small platforms, fixed interfaces are usually assigned as being in slot 0. In high-end platforms, physical slots exist to hold a Flexible PIC Concentrator (FPC), which is a large board that can in turn hold many interface cards.
- *PIC:* the second number corresponds to the Physical Interface Card (PIC) position within the slot.
- *port:* the third number corresponds to the port number on the PIC.

Let’s go back to the example cited previously. The interface name of *ge-0/0/1*. Here, the type of interface is *Gigabit Ethernet*, the slot number is 0, the PIC number is 0, and the port number is 1.

TIP The port number is also written on the PIC itself, which is useful when you're juggling multiple FPCs and PICs.

Try It Yourself: view our hardware configuration

Try entering the following operational mode command on your lab device to determine its physical configuration:

```
root@juniper1> show chassis hardware
```

Logical Units

In setting up your network you may want to partition a physical interface into multiple logical interfaces—for instance, subdividing an Ethernet interface into multiple virtual LANs (VLANs). The Junos OS refers to these logical interfaces as *units*. Typically Junos OS requires that you set one (or more) logical units on each physical interface.

In naming the logical interface, Junos OS simply appends the logical unit to the physical port. If you added a logical unit (also sometimes referred to as a *channel*) of 0 to the example above, the complete interface name would become: *ge-0/0/1.0*.

NOTE Some network vendors refer to logical interfaces on their platforms as *sub-interfaces*.

Gigabit Ethernet

The best way to learn how to configure interfaces in Junos OS is to present an example. Let's say that you want to configure the Gigabit Ethernet interface ge-0/0/1. Use the set interfaces command, specifying the IPv4 address 192.168.100.1/30:

```
root@juniper1# set interfaces ge-0/0/1 unit 0 family inet address 192.168.100.1/30
```

Looking at the levels of the command more closely:

- `ge-0/0/1` is the name of the Gigabit Ethernet physical interface.
- `unit 0` is a logical unit configured within the physical interface. Each physical interface must have at least one configured logical interface, with the first one numbered 0 (not 1) before it can carry traffic.

- `family inet` identifies the protocol used by the logical interface. You almost always want to configure at least one family on each logical interface. In this book, all of our configurations use *inet* which is how Junos OS refers to IPv4.
- `address 192.168.100.1/30` is the address of the logical interface. Each logical interface can support multiple addresses. So configuring additional addresses does not override existing addresses.

ALERT! In the configuration hierarchy, the software considers any statements configured directly under the interface name (for example [`edit interfaces ge-0/0/1`]) to apply to the entire physical interface. Similarly, statements configured directly under the unit number are all properties of the particular logical interface.

VERIFY Show that the Gigabit Ethernet interface has been set up:

```
root@juniper1> show interfaces
ge-0/0/1 {
         unit 0 {
    family inet {
      address 192.168.100.1/30;
           }
  }
}
```

Reviewing Your Work

With the essential basics now set up in your device, you may want to review your work.

First, commit your configuration.

Now you can use the `show configuration` command in operational mode to see what's in your active configuration after completing the commit. If you've been following along on your own device, your configuration should include the following statements for the ge-0/0/1 interface:

```
root@juniper1> show configuration interfaces ge-0/0/1
        unit 0 {
            family inet {
                address 192.168.100.1/30;
            }
        }
```

How to display the candidate as a series of set commands:

Here's a really good tip. You can also easily convert the displayed listing into the original set commands by piping the output into the `| display set` modifier. This makes it easy for you to see which commands created the configuration, and then to review or proof your work:

```
root@juniper1> show configuration interfaces ge-0/0/1 | display set
set interfaces ge-0/0/1 unit 0 family inet address 192.168.100.1/30
```

NOTE The Junos OS displays the `show configuration | display set` listing from the top of the configuration mode, therefore, the set commands are in the form you would use at the [edit] hierarchy level of the configuration.

TIP When looking at output with operational mode show commands, you can display more information by using the `detail` or `extensive` keywords.

Chapter 7

Setting Up User Accounts

Don’t leave yet! Get back in the lab and let’s continue learning device set up basics like how to configure user accounts and provide messages to users, as well as how to set up remote authentication and enable remote access.

The Junos OS offers a rich and flexible set of features for configuring and managing user accounts, authentications, and permissions. This chapter shows you what you need to get up and running on day one, but also requires a few advanced capabilities to help you get set up.

Creating Login Banners

You can create login banners that post messages and announcements to those who access the device. Let’s create an initial login message, now, before creating the first user accounts, so they’ll see it upon login.

Login Message

Login messages display a banner to all users when they access the device, before they log in. The message can be split over multiple lines by using \n, or *newline* (equivalent to a carriage return and line feed) as a delimiter:

```
root@juniper1# set system login message “ Welcome \n to the\n Junos OS Training\n “’
```

After you set up the message and commit the configuration, your users accessing your device will see it displayed on their screen. For example, if the remote client is using SSH:

```
$ ssh juniper1
 Welcome
 to the
 Junos OS Training
root@juniper1’s password:
```

TIP Use the login message to warn that unauthorized access to your device is prohibited (ask your legal department for the preferred statement in your organization), but here’s a sample:

```
root@juniper1# set system login message “WARNING: Unauthorized access is NOT Allowed.”
```

Login Announcement

Sometimes you want to make announcements only to authorized users after they have logged in. For example, use the `set system login announcement` command to announce an upcoming maintenance window to authorized users:

```
root@juniper1# set system login announcement “Maintenance scheduled 11PM to 2AM
tonight”
```

Configuring Login Accounts

The Junos OS requires that all users have a predefined account before they can log in to the device. Further, you can configure the login accounts to restrict who has access to what on your device.

The accounts can be set up with a local user and password, or as local users and user templates that depend upon remote servers to provide authentication either using the RADIUS or TACACS+ protocols. The rest of this chapter focuses on these two options for setting up your lab’s login accounts.

Local User and Password

Set up your local users with a name and password in the following steps, and then add their user class in the next following section.

How to set up local user accounts:

Set up as many local users as you need with the following steps:

1. To begin setup, navigate to the `[edit system login]` section of the configuration:

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

2. Add a new user using their assigned account login name. This example creates a new user with username *jadmin*:

```
[edit system login]
root@juniper1# edit user jadmin
```

3. You can also configure a full descriptive name for the account. If the full name includes spaces, enclose the entire name in quotes:

```
[edit system login user jadmin]
root@juniper1# set full-name “Juniper Network Administrator”
```

4. Set the user identifier (UID) for the account. As with UNIX systems, the UID enforces user permissions and file access. If you don’t set the UID yourself, the software will assign one for you. The format of the UID is a number in the range of 100 to 64000. This example shows how to set a UID:

```
[edit system login user jadmin]
root@juniper1# set uid 1250
```

5. Create a password for the user. As discussed in Chapter 6, you use the `set` command to create a password as plain text, and the Junos OS internally encrypts it:

```
[edit system login user jadmin]
root@juniper1# set authentication plain-text-password
New password: ####
Retype new password: ####
```

By default, the Junos OS locally authenticates all users who try to log in to the software using the accounts provided in the configuration.

If you try to commit the user configuration you have created so far, you will get an error that you are missing a mandatory statement: *the user class*. The user login class is important enough to deserve a section of its own, which follows.

MORE? The user (and root) passwords can also be locally configured as encrypted passwords. Find out how to set these up in the Configuring User Accounts section of the *Getting Started Guide for Routing Devices* in the Juniper TechLibrary: http://www.juniper.net/techpubs/en_US/junos15.1/topics/concept/security-user-account-understanding.html.

Login Classes

In addition to the user name and password, all user accounts require configuration of a login class. The login class defines the permissions for executable commands. As users enter commands in the command line, the Junos OS checks the login class permission level for each command before accepting it.

The Junos OS comes with four pre-defined user login classes:

- *superuser:* all permissions
- *operator:* clear, network, reset, trace, and view permissions
- *read-only:* view permissions
- *unauthorized:* no permissions

For the *jadmin* user created in this example, let's set the login class as *super-user*. You should always have at least one super-user set up locally in the device:

```
[edit system login user jadmin]
root@juniper1# set class super-user
```

How to set up custom login classes:

If you need additional permissions beyond the four default classes, you can create your own custom login classes. You can specify exactly which commands you want to include or exclude for each custom login class. In this way, you can create user classes tailored to the specific needs of each particular user group.

1. Create a custom login class just for network operations staff that you call *netops*:

```
root@juniper1# set system login class netops
```

2. For each login class you can specify which permissions you want to allow or deny, but here let's keep things simple by giving the *netops* class access to everything:

```
root@juniper1# set system login class netops permissions all
```

MORE? To find out more about user classes, including how you can set up your own user classes, see the Juniper TechLibrary at: http://www.juniper.net/techpubs/en_US/junos15.1/topics/task/configuration/access-login-class.html.

Setting Up Remote Authentication

It's common practice to use remote authentication servers to centrally store information about users (see Chapter 5). You can configure the Junos OS to use one or more remote authentication servers, including RADIUS and TACACS+ servers.

To set up remote authentication in your device, you need to configure the access to the server, the authentication order, and the local user accounts. Several options are available for mapping users authenticated by remote servers to the locally-defined user accounts of the device.

The following examples include a best practice method of using the remote template account. The *username remote* is a special term in the Junos OS. It acts as a template for users who are authenticated by a remote RADIUS or TACACS+ server, but do not have a locally-configured user account on the device. In this way, the software applies the permissions of the remote template to those authenticated users without a locally-defined account. All users mapped to the remote template are of the same login class.

NOTE Another method for mapping remotely authenticated users is to set up a common shared account for all users of the same user class. Use this method when you need more than one type of template for remote users. See the Juniper TechLibrary at https://www.juniper.net/techpubs/en_US/junos15.1/topics/task/configuration/authentication-user-local-template-account-configuring.html.

How to configure authentication by a RADIUS server:

Use the following steps to set up user authentication by a RADIUS server.

1. Enter the RADIUS configuration statement:

```
root@juniper1# set system radius-server 172.26.27.5
```

2. You can also include a shared secret in the command statement and, if necessary, the port number:

```
root@juniper1# set system radius-server 172.26.27.5 port 1845
root@juniper1# set system radius-server 172.26.27.5 secret Jun1p3r
```

3. Set the authentication order to include RADIUS:

```
root@juniper1# set system authentication-order radius
```

ALERT! Setting the system authentication without the password option (see the following section on Combining Login Methods) allows *only* RADIUS authentication, unless the RADIUS server is unreachable. In this example, the Junos OS only grants access if the RADIUS server authenticates the user. It does not attempt to use local passwords. However, if (and only if) the RADIUS server is not reachable, the software defaults to trying its own local password database.

4. Each user needs a locally defined username (such as *adminjlk* in this example), or you can establish the default remote user. If a given authenticated user name is not found locally on the device, then it defaults to the settings of the remote template:

```
root@juniper1# set system login user adminjlk class super-user
root@juniper1# set system login user remote class super-user
```

VERIFY If all is correct on the server, you should see the following messages in the syslog message file. Note that in order to use this verification you need to first configure system logging, as described in Chapter 8:

```
root@juniper1> show log messages
Apr 22 13:38:58  juniper1 sshd[17859]: Accepted password for adminjlk from 172.30.48.10
port 61729 ssh2
```

VERIFY Show the SSH session connections:

```
root@juniper1> show system connections
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address            Foreign Address       (state)
tcp4       0     48  172.30.53.101.22         172.30.48.10.61729     ESTABLISHED
```

If the user has no login on the RADIUS server, the message logs include the error message:

```
Apr 22 13:40:57  juniper1 sshd[17873]: Failed password for username from 172.30.48.10
port 64844 ssh2
```

If there's an issue on the RADIUS server, the messages file in Junos reports it. Here it's seen after the issue was found, and the user authenticates locally (passing the login criteria):

```
root@juniper1> file show /var/log/messages
Apr 22 13:44:39  juniper1 sshd: rad_send_request: No valid RADIUS responses received
Apr 22 13:44:39  juniper1 sshd: detected authentication server problem
Apr 22 13:44:39  juniper1 sshd: will attempt local password authentication
Apr 22 13:44:39  juniper1 sshd: local password authentication of user ‘jadmin’,
succeeded
Apr 22 13:44:39  juniper1 sshd[17893]: Accepted password for jadmin from 172.30.48.10
port 54817 ssh2
```

How to configure authentication by a TACACS+ server:

The following steps set up user authentication by a TACACS+ server.

1. Enter the TACACS+ configuration statement:

```
root@juniper1# set system tacplus-server 172.26.27.6
```

2. You can also include a shared secret, and if necessary the port number, in the command statement:

```
root@juniper1# set system tacplus-server 172.26.27.6 port 49
root@juniper1# set system tacplus-server 172.26.27.6 secret Jun1p3r
```

3. Set the authentication-order to include TACACS+:

```
root@juniper1# set system authentication-order tacplus
```

ALERT! Similar to the preceding RADIUS example, setting the system authentication without the password option allows only TACACS+ authentication, unless the TACACS+ server is unreachable.

4. Repeating the same steps from the RADIUS example, the following sets up the local user accounts. Each user needs either a locally-defined username (such as *adminjlk* in this example), or you can establish the default remote user:

```
root@juniper1# set system login user adminjlk class super-user
root@juniper1# set system login user remote class super-user
```

5. Verify if everything is correct on the server. You should see the following messages in the syslog message file (to use this verification you need to first configure system logging, described in Chapter 8):

```
root@juniper1> show log messages
Apr 21 16:36:40  juniper1 sshd[22387]: Accepted password for adminjlk from 172.26.24.4
port 33445 ssh2
```

6. Verify and view the SSH session and TACACS+ connections:

```
root@juniper1> show system connections
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address       Foreign Address        (state)
tcp4       0      0  172.26.27.44.58234  172.26.27.4.49        TIME_WAIT
tcp4       0      0  172.26.27.44.22     172.26.24.4.33445     ESTABLISHED
```

How to combine login methods:

Typically, you might want to combine multiple login methods and establish a sequential authentication order:

```
root@juniper1# set system authentication-order [ radius tacplus password ]
```

In this example, when the user logs in, the Junos OS performs these sequential steps:

1. Attempts to authenticate against the RADIUS server. If successful, the user is logged in.

2. If the first step fails, the software attempts to authenticate with the TACACS+ server. If this is successful, the user is logged in.

3. If the second step also fails, the Junos OS attempts to authenticate with the local password configured on the device. If this is successful, the user is logged in. If not, the device rejects the user login attempt.

Enabling Remote Access

SSH, Telnet, and FTP are widely used standards for logging in to network devices and exchanging files between systems, so that you don't always have to be physically at the console port. Before authorized users can access your device, or your device can exchange data with other systems, you must configure one or more of these enabling services. They are all *disabled* by default in the Junos OS.

SSH is Telnet's successor and is the recommended method for remote access. SSH encrypts all traffic, including passwords, so it can effectively eliminate eavesdropping, connection hijacking, and other attacks. The SSH utility includes SCP (Secure Copy Protocol), a file transfer program that uses SSH and is the recommended method for secure file exchange.

Use the following commands to set up the services that are needed in your device:

```
root@juniper1# set system services ftp
root@juniper1# set system services telnet
root@juniper1# set system services ssh
```

BEST PRACTICE Since both Telnet and FTP are legacy applications that use clear text passwords (therefore creating a potential security vulnerability), it's recommended that you use SSH and SCP. If you don't intend to use FTP or Telnet, it's not required to configure them on your device. However, don't forget to consider that some users may use FTP to store configuration templates, retrieve software, or other administrative tasks.

DON'T FORGET Commit your work before leaving this chapter or your lab, so it will become part of the active configuration.

Chapter 8

Configuring System Logs

System logs (often called *syslogs*) are UNIX-style distributed event reporting mechanisms that provide an extremely flexible method of generating and handling event messages with a higher degree of verbosity than traditional SNMP traps. Syslog was traditionally used in UNIX environments, with distributed hosts being configured to forward their messages to one central syslog server called the *log host*.

Syslog messages are categorized by a facility to indicate the source of the event and a level to indicate its severity. The Junos OS can respond to each event with a variety of actions based upon a combination of the facility and level reported in the message. These actions include displaying event messages locally, storing the messages locally, or forwarding messages to a standard remote syslog server for real-time or offline analysis by third-party monitoring applications. Administrators can send either all or selected syslog event messages to a central server for default duplication and analysis.

TIP You can find out what syslog configuration options are supported by the version of the Junos OS you are running in your device by entering the following operational command:

```
jadmin@juniper1> help reference system syslog
```

Syslog Destinations

The Junos OS offers the flexibility to send syslog messages to a variety of different destinations, as shown in Figure 8.1.

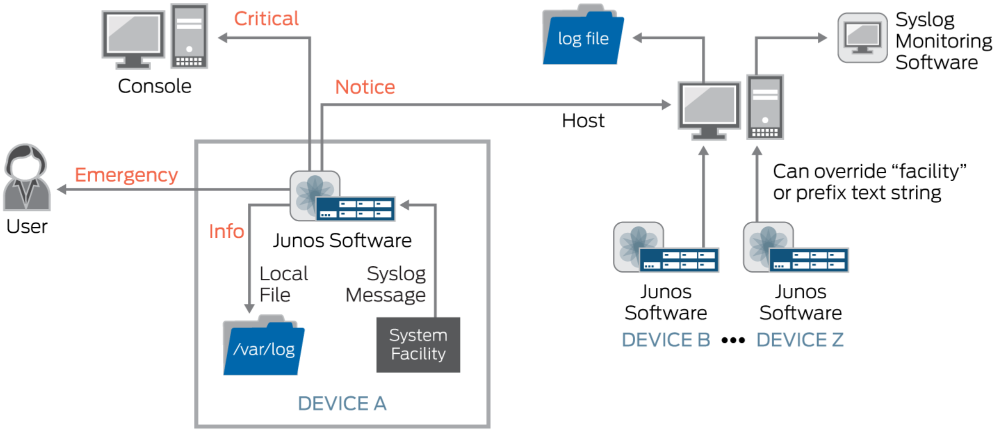


Figure 8.1 Syslog Destinations

The detailed descriptions of the possible destinations for log messages shown in Figure 8.1 are listed in Table 8.1.

Table 8.1 Syslog Destination Detail

Destination	Description
console	Display log messages on the device's console.
file	Store log messages locally on the device's hard disk. Typically log files are stored under the /var/log directory.
host	Forward log messages on to another syslog server (typically another Junos device or a UNIX machine) for additional processing.
user	Display log messages to a user on whichever pty (pseudo-teletype) terminal they are logged in on.

You can forward messages to a combination of all four destinations, as incorporated in the examples provided in this book.

Message Facilities and Levels

When setting up the system logging, you can also choose which messages to log, which facilities (sources) to use, and which levels (severities) are important to your network.

The Junos OS supports a number of syslog facilities to specify the software process generating the message. If you are unsure of which facility you want to monitor, or if you don't know which facilities are important, you can monitor them all and adjust them later!

Each facility can generate a number of different messages, each of a varying level or severity. For example, an informational message denotes the occurrence of a non-critical event, which does not require immediate action, whereas a critical event from the same facility might require immediate operator intervention. The full list of syslog levels supported by the Junos OS is shown in Table 8.2.

When you set the syslog level, you are defining a mask that specifies the lowest level of log message that you want to handle. For example, if you specify that you want to capture all log messages of warning level, the software logs messages to that level *and above*, notifying you of any error, critical, alert, or emergency level messages.

Table 8.2 Syslog Levels

Numerical Level	Level	Description
-	none	Disables logging of the associated facility to a destination.
0	emergency	System panic or other condition that causes the software component to stop functioning.
1	alert	Conditions that require immediate correction, such as a corrupted system database.
2	critical	Critical conditions, such as physical errors.
3	error	Error conditions that generally have less serious consequences than events in the emergency, alert, and critical levels.
4	warning	Conditions that warrant monitoring.
5	notice	Conditions that are not errors but might warrant special handling.
6	info	Events or non-error conditions of interest.
7	any	All (i.e. any) level of message from the facility.

MORE? For a full list of all of the messages supported by the Junos OS, refer to the Juniper TechLibrary: https://www.juniper.net/techpubs/en_US/junos15.1/information-products/topic-collections/syslog-messages/preface.html.

Sending Messages to a File

Now that you are familiar with syslogs and what they do in the Junos OS, the simplest and the most useful syslog configuration for your *Day One* lab work is to send messages to a local file. Storing messages locally can be useful if you need to maintain an audit trail of messages, or if you want to store messages for analyzing at another time.

How to configure a syslog file in a local file:

1. Enter the `file` statement in the `[edit system syslog]` hierarchy of the Junos configuration:

```
jadmin@juniper1# edit system syslog file all-messages

[edit system syslog file all-messages]
jadmin@juniper1# set any warning
jadmin@juniper1# set authorization notice
```

This example creates a file called all-messages in the local directory `/var/log` that stores messages from any facility at the level of warning and above. It also stores authorization messages, such as user logins, at the level of *notice* and above. The Junos OS automatically rotates log files and archives them periodically as they grow larger.

NOTE You can customize the archive behavior by altering the `[system syslog archive]` setting in the configuration – although that is not necessarily a *Day One* task.

2. After you have committed the configuration, you can view the log messages that are stored in a local file in operational mode:

```
jadmin@juniper1> show log all-messages | last 10
```

In this example the `last` command modifier requests only the last 10 lines (the most recent 10 events) in the file `all-messages` to show on the screen.

Directing Messages to a Terminal

When messages are important and require immediate intervention by a human operator, the best way to get the operator's attention is to display a message on their screen. The Junos OS allows you to display messages to a user if they are logged in and also to display messages on the device's console.

How to send messages:

1. Enter the user name in the `[system syslog user]` hierarchy:

```
jadmin@juniper1# set system syslog user * any emergency
jadmin@juniper1# set system syslog user jadmin any critical
```

The first set command forwards any emergency log message to all users, as indicated by the wildcard character < * >. The second set command forwards any critical level log message and above to the screen on which the user `jadmin` is logged in. This can be an effective way of forwarding important messages to administrators who are remote.

2. Another option is to forward log messages to the screen attached to the device's console. The following example sends log messages from any facility at the level of *error* and above to the device's console:

```
jadmin@juniper1# set system syslog console any error
```

Forwarding Messages to a Remote Server

As shown in Figure 8.1, the Junos OS can also forward syslog messages to one or more remote devices. Since many enterprises use fault monitoring software to receive and interpret syslog messages, it makes sense to configure your device to not only report log messages locally, but also to forward copies of each message to a remote fault monitoring system.

For example, suppose you have a device in your network named `loghost`, which is running a third party fault monitoring system. Simply specify the appropriate facilities and levels under the `[edit system syslog host]` hierarchy of the configuration:

```
jadmin@juniper1# set system syslog host loghost any notice
```

This example forwards log messages from all the facilities at the level of notice and above to the remote host called `loghost`.

Customizing Log Message Formats

You can customize log files in several ways. For example, you can include a match statement to exclude (with the < ! > character) specific logs using a regular expression. In this case, the log excludes the `RSVP INCORRECT FLOWSPEC` condition:

```
syslog {
    file messages {
        any warning;
        authorization notice;
        cron error;
        daemon info;
        kernel error;
        pfe info;
        match “!(.*RSVP_INCORRECT_FLOWSPEC.*)”;
        archive size 1m files 20 world-readable;
        explicit-priority;
    }
}
```

Many enterprises use commercial fault monitoring software and these software packages are capable of scaling to handle the huge volume of syslog messages that all of the devices in an IP network can generate. Typically such products parse all incoming messages, filtering them through a rules-based engine in order to determine what actions to take. While these systems can be very flexible, they can also be complicated to configure.

Junos OS helps to make integration with third-party fault monitoring systems easier. You can customize log messages, thereby reducing the amount of configuration and processing that is necessary at the receiving end.

Prefixed Strings

Let’s assume that your network has over 500 IP devices, all of which are generating syslog messages. Some may be Junos devices, some may be UNIX servers, and some may be networking devices from other vendors. Although most commercial fault monitoring applications can parse syslog messages by using custom rules, creating vendor-specific rules can be difficult and time consuming. One option is to prefix all syslog messages that originate from Junos OS with a particular string, thereby making it easier to remotely match and parse those messages.

Use this command to prefix the word *Junos* to each syslog message before it is forwarded to the remote device loghost:

```
jadmin@juniper1# set system syslog host loghost log-prefix JUNOS
```

This example adds a static prefix of JUNOS to each syslog message from the device. The prefix now makes it much easier for a remote fault monitoring application to identify any log message that came from the device.

Including Priority Information

An event's facility and severity level are referred to as its *priority*, and while the priority is known to the syslog servers, it is not often visible in the message text itself. Because of this, it can be difficult to determine the priority of a given message, especially if the message is forwarded to a remote host or stored locally in a file along with messages of other priorities.

You can insert the event priority into the message text by including the `explicit-priority` statement in the `[system syslog host]` or `[system syslog file]` hierarchy:

```
jadmin@juniper1# set explicit-priority
```

The `explicit-priority` statement inserts the priority in the form of `facility-level` into the beginning of each message. Note that in this case, the level is a numerical value, as specified in Table 8.1.

Facility Override

Some third party syslog monitoring applications "listen" to messages arriving with a specific facility. Since the Junos OS can generate messages from many different facilities, some of which may be unknown to the remote system, it is sometimes useful to override the original value to ease integration.

To override the original syslog facility, use the `facility-override` statement when configuring the syslog host:

```
[edit system syslog host loghost]
jadmin@juniper1# set facility-override local7
```

BEST PRACTICE

In general, it makes sense to specify an alternate facility that is not already in use on the remote system, such as one of the "localX" facilities. On the remote machine, you must also configure the syslog application to handle the messages in the desired manner.

When you have finished configuring the syslog settings you can activate the new configuration by running the `commit` command:

```
[edit system syslog]
jadmin@juniper1# commit
```

MORE? Syslog is one of several mechanisms that log network events. The other commonly used tool is trace logging (also known as *trace options*). Trace logging keeps track of specific processes, such as routing packets sent and received. Trace option output is similar to debug output in other systems. To learn more about trace logging visit the Juniper TechLibrary at https://www.juniper.net/techpubs/en_US/junos15.1/topics/concept/junos-software-tracing-logging-operations-overview.html.

Reviewing Your Work

This chapter concludes the basic setup of your device. Chapter 9 provides you with additional commands and shortcuts that can ease configuration moves, adds, and changes in your device.

If you have been following along in your lab, and closely matching the past few chapters, your configuration should closely match the configuration listing provided in the Appendix.

Chapter 9

Working with Groups and Templates

This last chapter provides some handy techniques that can save you a lot of time when you are creating and modifying configurations in the Junos OS CLI. The techniques let you easily reuse configuration statements for other parts of your configuration and even in other devices. For example, you can use *configuration groups* to set up and apply common elements that are reused within the same configuration, and *configuration templates* to load common elements used in the configurations of different devices. These shortcuts cannot only speed configuration editing, but can also help reduce errors often associated with repetitive command entry.

Several of the examples might go beyond what you would typically do in a *Day One* lab, but consider it a sneak peek to many other powerful features within the Junos OS.

In many cases, the parameters established with apply groups are typed in ALL CAPS to make it easier to distinguish such custom configurations from the standard configuration parameters.

Defining Groups

For those settings repeated in many parts of the configuration, such as interface parameters, *configuration groups* let you streamline setup. Configuration groups are sets of statements that you can apply to multiple parts of the configuration to create smaller, more logically constructed configuration files. Not only is your initial setup faster, but when you need to make a change, you can do it in one place, and yet have it apply everywhere.

This section provides two examples of using groups in the configuration of interfaces.

It's important to remember that *where* you apply configuration groups within the configuration matters, as only that specific hierarchy level (and below) inherits the group statements. Further, the *ordering* of configuration groups is also important, as the Junos OS inherits statements in the order that they are applied.

Creating an Interface Group

Many wide area links are based on the optical SONET/SDH standard. All of your WAN interfaces will require SDH framing, and need to comply with parameters as defined by RFC 2615. Let's assume you have a SONET/SDH interface configured like this:

```
jadmin@juniper1# show interfaces so-0/0/0
unit 0 {
    family inet {
        address 192.168.1.1/30;
    }
}
```

How to configure and apply the interface group:

1. Configure the group and set the required parameters:

```
set groups SDH interfaces <so-*> framing sdh
set groups SDH interfaces <so-*> sonet-options rfc-2615
```

SHORTCUT The interfaces are configured as `<so-*>` which acts as a wildcard for all SONET/SDH interfaces, so when it is applied, all interfaces inherit these settings.

2. Now apply the group where you want it in the configuration. If you enter the `set apply-groups` command from the top of the configuration tree, the group is applied throughout the configuration (in this case, you could also apply the group at the interface level: `set interfaces apply-groups SDH`):

```
jadmin@juniper1# set apply-groups SDH
```

It’s important to note that the `show` command only displays configurations under the specific branch in which you are currently working. The listing does *not* show any settings *inherited* from any configuration groups applied in other parts of the configuration. For example, if you use the following show command, you do not see the SDH group, even though you applied it:

```
jadmin@juniper1# show interfaces so-0/0/0
unit 0 {
    family inet {
        address 192.168.1.1/30;
    }
}
```

Instead, pipe the `show` output through the `display inheritance` option to show the full configuration with the applied SDH group:

```
jadmin@juniper1# show interfaces so-0/0/0 | display inheritance
##
## ‘framing’ was inherited from group ‘SDH’
##
framing {
    ##
    ## ‘sdh’ was inherited from group ‘SDH’
    ##
    sdh;
}
##
## ‘sonet-options’ was inherited from group ‘SDH’
##
sonet-options {
    ##
    ## ‘rfc-2615’ was inherited from group ‘SDH’
    ##
    rfc-2615;
}
unit 0 {
    family inet {
        address 192.168.1.1/30;
    }
}
```

SHORTCUT Included comments may make the configuration difficult to read. Use the `except` command to hide the comments from the listing:

```
jadmin@juniper1# show interfaces so-0/0/0 | display inheritance | except ##
framing {
    SDH;
}
sonet-options {
    rfc-2615;
}
unit 0 {
    family inet {
        address 192.168.1.1/30;
    }
}
```

How to exclude an apply-group:

Now let's show you how to exclude a broadly-applied group from specific sections of your configuration.

Let's assume that your network uses the ISO and MPLS protocols in a group applied at the top of the configuration. Doing this means that you don't need to configure these families under each interface. The wildcard < * > notation is used to ensure the protocols are configured throughout:

```
groups {
        ISIS-MPLS {
        interfaces {
            <*-*> {
                unit <*> {
                    family iso;
                    family mpls;
                }
            }
        }
    }
}
apply-groups ISIS-MPLS;
```

Now, let's assume there are some interfaces where you don't want these protocols configured. For instance, you may not want to enable the ISO or MPLS protocols on interfaces within a Level 3 VPN.

Use the `apply-groups-except` statement to exclude a broadly-applied group within a specific part of the configuration:

```
set interfaces ge-0/0/1 apply-groups-except ISIS-MPLS
```

Using `apply-groups-except` tells the software to exclude the ISIS-MPLS group on the ge-0/0/1 interface, even though the ISIS-MPLS group has been applied at the top of the configuration.

MORE? For full details on configuration groups and applying them, see the Juniper TechLibrary: https://www.juniper.net/documentation/en_US/junos14.2/topics/concept/junos-software-configuration-groups-understanding.html.

Using Configuration Templates

Let's say you are working on a large network, where you are responsible for installing and configuring one hundred new Juniper Networks devices, with only a two-day window to finish the job. Because most of the devices operate in the same network, many of their configuration parameters are the same. Instead of typing configuration data into each device individually, it's possible for you to create a template configuration (full or partial) that you can copy to other devices. Using a template saves time and also reduces the risk of errors.

How to create a template:

The easiest way to create a template is to make a copy of an existing configuration or part of a configuration. Use the `save` command to save it into a file, and provide a file name as an argument:

```
[edit]
jadmin@juniper1# edit groups common

[edit groups common]
jadmin@juniper1# save common-template
Wrote 23 lines of configuration to ‘common-template’
```

This example creates a file called `common-template` that contains everything under the `[edit groups common]` hierarchy, including a timestamp and the opening groups statement. The file resides locally in the user's home directory within the device. In this case, it's */var/home/jadmin*.

TIP Saving the template to an FTP server makes it easier for other devices to access it.

How to load a template:

If you have saved the configuration template locally as a file, you can use the `load` command from the top of configuration mode to load it into the device's configuration:

```
[edit]
jadmin@juniper1# load merge common-template
load complete
```

In this example, the `load` command includes the `merge` argument, which tells the software to merge the current candidate configuration with the contents of the loaded file. The Junos OS adds the template statements, exactly as you saved them, to the `[edit groups common]` hierarchy location of the device configuration.

TIP This example assumes that the template is stored locally as a file called *common-template*. Alternatively, if the template was stored on a remote FTP server, you would enter its location as a URL:

```
jadmin@juniper1# load merge ftp://user:password@server/junos/templates/common-template
```

After loading the file, don't forget to commit the new configuration.

Different Ways to Save Your Configuration

While you are still in the lab, let's show you different examples of ways to save the candidate or the active configuration. It can be easy to forget which configuration files you are saving, so note the difference between *candidate* and *active*.

The commands in this section show you how to create a file of the entire configuration, or a portion of it, and then save that file locally or on other devices. Additionally, you can configure the Junos OS to automatically save the active configuration file at specific intervals, or upon every commit.

How to save a candidate file locally:

Every Junos OS user defined in the configuration has their own home directory within the device at: /var/home/*username*.

1. To save the candidate configuration into your user home directory, simply save to a filename in configuration mode:

```
jadmin@juniper1# save router-config
Wrote 206 lines of configuration to 'router-config'
```

2. You can also save the configuration as a series of "set" commands. To do this, `show` the configuration and pipe the result first through the `display set` command and then pipe that result into the `save` command:

```
jadmin@juniper1# show | display set | save router-config-set-format
Wrote 206 lines of configuration to 'router-config'
```

Check your home directory on the device using the `file list` command from operational mode:

```
jadmin@juniper1# router-config
```

Use the `file show` command to view the actual contents of your saved configuration file:

```
jadmin@juniper1# run file show router-config
<configuration file contents will be here>
```

How to save a portion of the candidate configuration:

Use the save command deeper in the configuration to save portions of the candidate configuration as command blocks. You can reuse these command blocks in other devices in your network. For example, you could use the same system login information for all the switches in your network:

```
[edit system login]
jadmin@juniper1# save system-login
Wrote 29 lines of configuration to ‘system-login’
```

How to save a configuration file remotely:

This example saves the entire candidate file to a remote server called *remot*, using SCP (secure copy) to transport it:

```
[edit]
jadmin@juniper1# save scp://jadmin@remot
The authenticity of host ‘remot (172.26.25.4)’ can’t be established.
RSA key fingerprint is 13:ff:78:8a:fd:38:8f:d8:94:5e:39:9f:60:eb:9b:b5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added ‘remo,172.26.25.4’ (RSA) to the list of known hosts.
jadmin@remot’s password:
tempfile                                      100% 4482     4.4KB/s   00:00
Wrote 270 lines of configuration to ‘scp://jadmin@remot’
```

If you want to save the active configuration, you can use the operational `mode file copy` command. Optionally, you can use a pipe to save the output of the operational `mode show` command. The following command lets you save the active configuration by creating a listing of the active configuration, then piping into a save statement to create a locally-stored file called *Tuesday-archive*:

```
jadmin@juniper1# run show configuration | save Tuesday-archive
Wrote 115 lines of configuration to ‘Tuesday-archive’
```

How to automate saving the active configuration:

Let’s say you are making a copy of your Junos OS archive every Tuesday by logging in and making the copy. The Junos OS can do this for you, not just on Tuesdays, but every day. You can also configure the OS to automatically save the latest active configuration file and transfer it to a remote host.

1. If you have already set up an archive host, or set of hosts, use these commands to specify a URL for each host that tells the Junos OS where to send the configuration:

```
jadmin@juniper1# set system archival configuration archive-sites ftp://
jadmin:password@remot/archives
```

2. Now configure how often (in number of seconds) you want the Junos OS to save the active configuration. You can specify any interval from 15 minutes (900 seconds) up to 48 hours (2880 seconds):

```
jadmin@juniper1# set system archival configuration transfer interval 1440
```

This configures the Junos OS to take the active configuration and send a copy to the FTP remote server in the directory archives every 1440 seconds (every 24 hours, or once per day).

How to automate saving the active configuration upon commit:

A best practice is to configure the Junos OS to archive the active configuration after every commit (therefore, every time the configuration has been changed).

1. Set up the location to send the saved active configuration file:

```
jadmin@juniper1# set system archival configuration archive-sites ftp://
jadmin:password@remo/archives
```

2. Configure it to transfer the active configuration after every commit:

```
jadmin@juniper1# set system archival configuration transfer-on-commit
```

Now when anyone commits a change on the device, a copy of the latest active configuration is transferred to the remote archive host for any purpose you want.

Loading Configurations

You can use the `load` command to insert saved configuration files into the candidate. You can load a complete, or a partial, configuration from a local file, a file on a remote machine, or from a terminal emulator's capture window. A variety of options also let you manage exactly how the Junos OS integrates the loaded file into your candidate configuration. Let's review.

load override

Use the `load override` command to completely replace the current candidate configuration with a previously stored file. You must enter the `load override` command from the top of the configuration mode.

This example loads the *router-config* file saved in the previous section to the */var/tmp* directory on the device, completely overwriting the existing configuration:

```
jadmin@juniper1# load override /var/tmp/router-config
load complete

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

REMEMBER Any newly loaded configuration file only replaces the candidate configuration. You must enter a `commit` command for it to become the active running file.

load merge

Instead of replacing a configuration, you may want to add a configuration snippet to your candidate configuration. You can use the `load merge` command to add the system login configuration statements saved previously in the local directory of the device:

```
[edit]
jadmin@juniper1# load merge system-login
load complete
```

This example loads the *system-login* file on the device, and merges it with the candidate configuration file from the top of the configuration tree. You must always enter the `load merge` command from the top of the configuration mode. The Junos OS adds these statements, as you save them, to the [edit system login] hierarchy location of your configuration.

The `save` command always captures the hierarchy reference from the root of the configuration, so the `load merge` command always adds the statements exactly in the same place as you saved them.

There may be times when you want to add saved statements to a different part of your configuration. See `relative` option discussed to see how to specify where the Junos OS loads the configuration statements of a saved file.

load merge terminal

Let's suppose that you want to copy the syslog settings that have already been configured on one device, and paste them onto another:

```
system {

   syslog {
       user * {
             any emergency;
        }
        host 172.26.27.8 {
             any notice;
             authorization info;
             interactive-commands info;
        }
        file messages {
              any notice;
              authorization info;
        }
    }
}
```

First copy the snippet from the source, using a copy command on your terminal, such as Control+C. Then enter the `load merge terminal` command on the destination router, and paste the snippet in on your terminal, for example, by using a paste command, Control+V:

```
[edit]
jadmin@juniper1# load merge terminal
[Type ^D at a new line to end input]

system {

   syslog {
       user * {
             any emergency;
        }
        host 172.26.27.8 {
             any notice;
             authorization info;
             interactive-commands info;
        }
        file messages {
              any notice;
              authorization info;
        }
    }
}

^D
load complete
```

ALERT! When using a `terminal` command, make sure you end the terminal with Control+D (`^D`).

The new syslog statements are now ready to be applied to your configuration:

```
jadmin@juniper1# commit
```

load merge terminal relative

Perhaps you want to merge a configuration snippet part way down inside a branch of the Junos configuration tree. If so, you can append the `relative` keyword to the `load merge` command.

Let's say that you want to copy just the syslog host from the previous example. Copy the host details using a copy command, making sure you include the very last curly bracket (`}`):

```
system {

   syslog {
       user * {
             any emergency;
        }
        host 172.26.27.8 {
             any notice;
             authorization info;
             interactive-commands info;
        }
        file messages {
              any notice;
              authorization info;
        }
    }
}
```

On the destination device, navigate to the desired section of the configuration:

```
jadmin@juniper1# edit system syslog
[edit system syslog]
jadmin@juniper1#
```

Then issue the `load` command as before now with the addition of the `relative` keyword:

```
jadmin@juniper1# load merge terminal relative
[Type ^D at a new line to end input]
host 172.26.27.8 {
    any notice;
    authorization info;
    interactive-commands info;
}
^D
load complete
[edit system syslog]
```

TIP You can also use the `relative` option when loading a snippet of a configuration from a file. The format of the command is similar in form to this example: `load merge <filename> relative`.

MORE? Find additional examples of how you can use load commands in the Junos OS see the Juniper TechLibrary at: https://www.juniper.net/techpubs/en_US/junos15.1/information-products/pathway-pages/junos-cli/junos-cli.html.

Summary

You have now completed *Exploring the Junos CLI, Second Edition*. There are many books in the *Day One* library (www.juniper.net/dayone) and professionally-published books in conjunction with O'Reilly Media in the *Juniper Technical Library* (www.juniper.net/books).

A complete and thorough documentation suite on the Junos OS begins in the Juniper TechLibrary's *Junos OS CLI Guide* at: https://www.juniper.net/techpubs/en_US/junos15.1/information-products/pathway-pages/junos-cli/junos-cli.html. From there you can branch off into various Junos OS topics at your discretion.

Appendix

Configuration Listing

This section provides the configuration listing for all the statements that this book has helped you to configure on your device.

The resulting configuration listing on your device may also have additional statements associated with previously defined default or preconfigured settings. If you chose to set up your device with the custom settings specific to your network, then your output will include those specific configuration names, addresses, etc.

```
## Last commit: 2015-06-16 08:32:35 CEST by root
version “14.2I0 [builder]”;
groups {
             ISIS-MPLS {
        interfaces {
            <*-*> {
                unit <*> {
                    family iso;
                    family mpls;
                }
            }
        }
    }
}
apply-groups ISIS-MPLS;
system {
    host-name juniper1;
    domain-name enterprise.com;
    domain-search [ enterprise.com department.enterprise.com ];
    backup-router 172.26.31.1 destination [ 172.26.31.1/32 172.16.0.0/12 ];
    time-zone Europe/Amsterdam;
    authentication-order [ radius tacplus password ];
```

```
        name-server {
            172.26.27.2;
                   172.26.27.3;
        }
        radius-server {
            172.26.27.5 {
                port 1845;
                secret «$9$8.wx-b4aU.PQZG39pu1INdb»;
            }
        }
        tacplus-server {
            172.26.27.6 {
                port 49;
                secret «$9$KyEWXNs2aikP4oT39CuOLxN»;
            }
        }
        login {
            announcement «Maintenance scheduled 11PM to 2AM tonight»;
            message «Welcome \n to \n JUNOS\n»;
            user jadmin {
                full-name “Juniper Network Administrator”;
                uid 1250;
                class super-user;
                authentication {
                    encrypted-password “$1$jetUXT44$D9KVQKofqwKMEfcBjp3zg0”;
                }
            }

            user remote {
                uid 2001;
                class super-user;
            }
            user adminjlk {
                uid 2002;
                class super-user;
            }
        }
        services {
            ftp;
            ssh root-login deny;
            telnet;
        }
        syslog {
            user * {
                any emergency;
            }
            user jadmin {
                any critical;
            }
            host loghost {
                any notice;
                facility-override local7;
            log-prefix JUNOS;
        }
        host set {
            explicit-priority;
        }
        file all_messages {
            any warning;
            authorization notice;
        }
        console {
            any error;
        }
        time-format;
    }
    ntp {
        boot-server 172.26.27.4;
        server 172.26.27.4;
    }
}
interfaces {
    ge-0/0/1 {
        apply-groups-except ISIS-MPLS;
        unit 0 {
            family inet {
                address 192.168.100.1/30;
            }
        }
    }
    fxp0 {
        unit 0 {
            family inet {
                address 172.26.27.44/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.26.0.110 {
                    preferred;
                }
                      address 127.0.0.1/32;
            }
        }
    }

}
```

Recommended Reading

Try any of these *Day One* books as your next step toward Junos OS proficiency at http://www.juniper.net/dayone.

- *Day One: Routing the Internet Protocol*
- *Day One: Junos for IOS Engineers*
- *This Week: Hardening Junos Devices, 2nd Edition*
- *Day One: Configuration EX Series Ethernet Switches, 3rd Edition*

Your Roadmap to Juniper Knowledge Poster

Use this poster as an informational roadmap to all the documentation, training, certification, and configuration resources available to you from Juniper Networks: http://www.juniper.net/posters.

Other Books by Walter Goralski

You might enjoy these other books by Walter Goralski:

- *Junos OS for Dummies, 2nd Edition* (http://www.amazon.com/JUNOS-OS-Dummies-Walter-Goralski/dp/0470891890.)
- *The Illustrated Network: How TCP/IP Works in a Modern Network* (The Morgan Kaufmann Series in Networking) 1st Edition (http://www.amazon.com/Illustrated-Network-Modern-Kaufmann-Networking/dp/0123745411.)
- *SONET/SDH 3rd Edition* (http://www.amazon.com/Sonet-SDH-Third-Walter-Goralski/dp/0072225246.)
- *Juniper and Cisco Routing: Policy and Protocols for Multivendor IP Networks*, Wiley (http://www.amazon.com/Juniper-Cisco-Routing-Protocols-Multivendor/dp/0471215929)
- *Optical Networking & WDM (Standards & Protocols)* (http://www.amazon.com/Optical-Networking-Wdm-Standards-Protocols/dp/0072130784.)