Junos® Fundamentals Series

DAY ONE: CONFIGURING JUNOS BASICS

It's day one and you have a job to do. With this Day One book you're just a few hours away from setup of the base system settings of your router, switch, or security device. Start configuring Junos today.

By Sean Clarke

DAY ONE: CONFIGURING JUNOS BASICS

This second book in the Junos Fundamentals Series helps you to configure the basic settings of your device and to learn more about configuration mode. These settings are the first steps to configuring a Junos device, whether you are setting up a router, switch, or security platform. It's fast and it's easy with step-by-step instructions. *Day One: Configuring Junos Basics* continues the practical tutorial for first-time users of Junos and Juniper products, but can also be used as a reference or refresher for more experienced Junos administrators.

"The Day One series is extremely useful in my day-to-day activities with Juniper devices. It answers my questions about how the Junos configuration process works and how to set up devices in my network, providing lots of practical tips and examples from Juniper experts. This is exactly what I was looking for."

Gadde Pradeep, JNCIA-M, JNCIS-M, JNCIA-EX, JNCIA-ER, JNCIS-ER

IT'S DAY ONE AND YOU HAVE A JOB TO DO, SO LEARN HOW TO:

- Create a handy checklist of settings to use in configuring the system basics
- Configure base system settings
- Create login accounts and permissions
- Set up SNMP to work with your existing systems
- Monitor your device remotely and configure system logs
- Installs Web-based management
- Make changes faster with configuration shortcuts
- Streamline device setup with configuration groups and templates
- Compare your resulting configuration to the booklet example

Juniper Networks Day One books provide just the information you need to know on day one. That's because they are written by subject matter experts who specialize in getting networks up and running. Visit www.juniper.net/dayone to peruse the complete library.

Published by Juniper Networks Books

ISBN 978-1-936779-02-4

51400

9 781936 779024

7100 1050

JUNIPer
NETWORKS

JUNOS® Software Fundamentals Series

Day One: Configuring JUNOS Basics

By Ian Jarrett and Sean Clarke

Juniper® NETWORKS

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice. Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Published by Juniper Networks Books
Writers: Sean Clarke
Editor in Chief: Patrick Ames
Copyediting and Proofing: Nancy Koerbel
Junos Program Manager: Cathy Gadecki

ISBN: 978-1-936779-02-4 (print)
Printed in the USA by Vervante Corporation.
ISBN: 978-1-936779-03-1 (ebook)

Version History: v4 January 2011
5 6 7 8 9 10 #7100105

About the Authors

Sean Clarke is a Senior Systems Engineer at Juniper Networks, with over nine years of hands-on experience with Junos.

Authors' Acknowledgments

The authors want to thank the many people who assisted us in creating this book. Our literary manager Patrick Ames guided us and Nancy Koerbel edited our writing. The Day One Series Editor Cathy Gadecki saved this project several times and tirelessly devoted many days to working with us. Michael Scruggs and Marilyn Kerr fine-tuned our attempts at instructional clarity. David Nguyen, Jerish Parapurath, Jennifer Pulsifer, and Brad Woodberg answered our questions and provided technical review of sections.

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

Send your suggestions, comments, and critiques by email to dayone@juniper.net.

Follow the Day One series on Twitter: @Day1Junos

Welcome to Day One

Day One booklets help you to quickly get started in a new topic with just the information that you need on day one. The Day One series covers the essentials with straightforward explanations, step-by-step instructions, and practical examples that are easy to follow, while also providing lots of references on where to learn more.

Why Day One Booklets?

It's a simple premise – you want to use your Juniper equipment as quickly and effectively as possible. You don't have the time to read through a lot of different documents. You may not even know where to start. All you want to know is what to do on the first day, day one.

Day One booklets let you learn from Juniper experts, so you not only find out how to run your device, but also where the shortcuts are, how to stay out of trouble, and what are best practices.

What This Booklet Offers You

Day One: Configuring JUNOS Basics helps you to configure the basic settings of your device and to learn more about configuration mode. These settings are the first steps to configuring a JUNOS device, whether you are setting up a router, a switch, or a security platform.

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

- ✓ Create a handy checklist of settings to use in configuring the system basics
- ✓ Configure basic system settings
- ✓ Create login accounts and permissions
- ✓ Set up SNMP to work with your existing systems
- ✓ Monitor your device remotely and configure system logs
- ✓ Install Web-based management
- ✓ Make changes faster with configuration shortcuts
- ✓ Streamline device setup with configuration groups and templates
- ✓ Compare your resulting configuration to the booklet example

Trying Things Out in Your Own Device

Having access to a JUNOS device while reading is useful for working along with the steps and commands of the examples. The Appendix available in the electronic version of this booklet provides a reference configuration. If you follow along in your own device by entering all of the configuration commands designated by an arrow ▶ in the left margin (and no other commands), your resulting configuration should match the expected configuration in the Appendix. For example:

```
▶ root@juniper1# set class super-user
```

To access JUNOS Software, you must first have access to the device itself, either through the console port or the management port. Because each network is different, the process of logging in to deployed equipment is beyond the scope of this booklet. So before attempting to log in to the JUNOS command-line interface, you need to know how to access your device on the network, or have physical access to it.

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

What to Enter into Your Device

With a focus on doing, Day One booklets include lots of examples. The examples show screen output in a `fixed-width` font with the commands that you enter in **`boldface`**. If the command includes user named input (such as the name of a file, group, or policy), the examples show it as ***`boldface italic`***. The booklet uses these conventions in both the examples and their discussion to identify types of input in the booklet. However, when you type commands into the command-line interface (CLI), all input is plain-text entry.

What You Need to Know Before Reading

Before reading this booklet, you should be familiar with basic networking concepts and the topics covered in the first booklet of the JUNOS Software Fundamentals series. *Day One: Exploring the JUNOS CLI* introduced the basic commands used to configure and operate the JUNOS device, summarized in the table in the Appendix (available in the electronic version).

Chapter 1

Creating a Checklist

JUNOS Software is a network operating system that runs many of the Juniper Network platforms. Whether it is your first day setting up a new router, a switch, or a security platform, this booklet walks you through the steps to configure the JUNOS basics. These steps are straightforward and follow common practices well-known to experienced network administrators.

As with any network device, regardless of the manufacturer, there are a number of fundamental settings to configure before you can declare it “ready for use.” For example, the device needs to know its assigned IP addresses, which users are allowed to enter, as well as its management basics.

This first chapter steps you through the list of information you may need to configure the device basics. Record this information in the margins or in a worksheet provided in the Appendix, so you have a record of vital information and because sometimes a pen and paper is easier to use when you’re moving among the racks. If you gather all the details described in the sections of this chapter, you can breeze through the basic device setup discussed in Chapters 2 through 5.

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 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 servers so that administrators can connect to the device using the 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. If your device is an EX Series Ethernet Switch or an M Series or MX Series router, it has a dedicated management port on the front panel. For other types of platforms you can configure a management interface on one of the network interfaces; this interface can be dedicated to management or shared with other traffic.

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 JUNOS devices (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 JUNOS 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 JUNOS, 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 rather than numbers especially if those numbers are IPv4 addresses. Because of this, domain name system (DNS) servers are used to map device hostnames to IP addresses and vice versa.

DNS allows you to 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 defined 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 use NTP server(s) on your network, ensure you know the address(es).

User Names and Passwords

The root account of JUNOS Software 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*.

You should keep all your passwords secret at all times. If you write down a password here, make sure that this document is kept safe and secure

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.

JUNOS helps to enforce the use of strong passwords. For example, valid passwords must:

- Be a minimum length of 6 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 the server(s) allow 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 Control Access Control System). JUNOS can be configured to query multiple remote authentication servers of both types.

MORE? If you want more information about RADIUS or TACACS+ technologies, see the *System Basics Configuration Guide* of the JUNOS technical documentation. All Juniper technical documentation is available at www.juniper.net/techpubs.

Network Interfaces

If your device is performing Layer 3 forwarding (e.g. 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 booklet helps you to configure a Gigabit Ethernet interface so that you can get your device “up and running” on your network.

MORE? *Day One: Configuring JUNOS Interfaces and Routing* discusses how to configure other types of interfaces. Check availability and download your copy at www.juniper.net/dayone.

Network Management System

Do you use a central network management system (NMS)? Most NMS's use a version of Simple Network Management Protocol (SNMP) that can monitor the status of JUNOS devices that send unsolicited messages called traps. You can configure the IP address of your NMS so that JUNOS knows where to send its traps.

SNMP uses a very basic form of authentication called community strings to control access between a manager and remote agents, and vice versa. Community strings are administrative names used to group collections of devices (and the agents running on them) into common management domains. If a manager and an agent share the same community, they can talk to one another.

Many people associate SNMP community strings with passwords and keys because the jobs they do are similar. As a result, SNMP communities are traditionally referred to as strings. The community string is the first level of management authentication implemented by the SNMP agent in JUNOS.

You may also want to configure remote logging on your device. JUNOS 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 day one configuring. Print out the Configuration Information Worksheet from the Appendix (included in the electronic version of this booklet) and record the information you've gathered. Or just write the details for your devices in the margins of a printed copy of this booklet. Now, you have a reference to complete the commands and steps described in the remaining chapters of this booklet.

Chapter 2

Configuring the System Basics

In the previous chapter you gathered the essential configuration information to use with this booklet. In this chapter, you begin the actual setup of your device with the basic settings, including the base system, user accounts, remote access, and interfaces.

Follow along in your own device by entering all the configuration commands designated by an arrow ▶ in the left margin. You can then compare your resulting configuration with the expected configuration included in the Appendix.

A pratical change is to customize your command entries using the specific information gathered in Chapter 1. When you compare your resulting configuration with the expected configuration in the Appendix, the only differences will be in the customizable fields, and your device will be ready with the basic settings to run in *your* network.

Configuring Base System Settings

This section guides you through the first steps in configuring your device, including base system settings of root (administrator) password, hostname, management interface, loopback interface, and backup router. Follow along and refer to the information that you gathered in Chapter 1 to customize the setup of your device for your network.

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

TIP If you are new to JUNOS, or it has been awhile since you last configured a new device, explore the `[edit system]` hierarchy as a reminder of 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 JUNOS. The root user is by default the administrator or super user, who has absolute permission to both configure and install software on a device.

JUNOS 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.

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

As you enter the password in plain text, JUNOS encrypts it immediately. You don't have to tell JUNOS to encrypt the password as in some other systems. Plain text passwords are therefore hidden and marked as ## SECRET-DATA in JUNOS configuration listings (see Appendix example).

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

```
set system services ssh root-login deny
```

Hostname

The hostname of the device provides its identification for many purposes. JUNOS 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. In this booklet we use the name ***juniper1***, but you can choose something more descriptive:

```
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.

JUNOS follows the IP convention of using `lo0` as the loopback interface’s identifier name. Refer to Chapter 1 as a reminder of what you have chosen as the IP address of your loopback interface:

```
set interfaces lo0 unit 0 family inet address 192.26.0.110/32
```

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

ALERT! JUNOS 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:

```
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:

```
set interfaces lo0 unit 0 family inet address 192.168.1.1/32
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:

```
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:

```
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 (set up in Chapter 3).

Many types of JUNOS platforms include a dedicated management port on the front panel. For others, you can configure one of the Ethernet ports to act as the management interface.

Platforms that use a network Ethernet interface for management include the SRX100, SRX210, SRX240, and SRX650 Services Gateways and the J Series Services Routers. A network interface can be configured as being dedicated to out-of-band management or as being shared by both management *and* network traffic.

MORE? 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. Find out how to configure the EX Series Ethernet Switches with in-band management by downloading the white paper *Deploying EX-series Switches in Branch Offices* at www.juniper.net/us/en/products-services/switching/ex-series/.

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 the 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 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:

```
▶ set interfaces me0 unit 0 family inet address 172.26.27.44/24
```

On other JUNOS devices the dedicated management port is named fxp0. Table 2.1 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 2.1 **Names of Dedicated Management Ports**

Platform	Dedicated Management Port
EX Series Ethernet Switches	me0 (see note at the top of next page)
M, MX, and T Series Routers	fxp0
SRX5xxx and SRX3xxx Services Gateways	fxp0

MORE? Turn to the J Series documemtation for more about its management interface as www.juniper.net/techpubs.

NOTE The EX Series also includes a commonly used virtual management interface known as vme that is used for managing devices grouped together in a virtual chassis. For more information check the availability of *Day One: Configuring EX Series Ethernet Switches* at www.juniper.net/dayone .

Management Over a Network Interface

If your type of device uses a network interface for carrying management traffic, you similarly need to configure it with the IP address that you want to use as the management interface. The following example sets up the management interface on a network interface which is dedicated to out-of-band management. The Appendix outlines how to set up management on a shared network interface.

The following section discusses how to configure the management interface on a branch SRX Series device or J Series in flow-based mode.

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

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

```
set interfaces ge-0/0/0 unit 0 family inet address
172.26.27.44/24
```

2. Before the interface can carry traffic, you must assign the configured management interface to a zone. The zone provides virtual separation of traffic and acts as a policy enforcement point. 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:

```
set security zones 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:

```
set security zones functional-zone management host-inbound-
traffic system-services ssh
```

MORE? Learn more about functional and security zones in the *Security Configuration Guide* available at www.juniper.net/techpubs.

Backup Router

You can configure JUNOS to use a backup router during the initial boot process in Layer 3 devices. The JUNOS process responsible for establishing routes (among other functions) is known as the routing protocol daemon (RPD). When JUNOS 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 during boot time, thereby minimizing the amount of time the device is unavailable.

```
set system backup-router 172.26.31.1 destination 172.16.0.0/12
```

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 JUNOS 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 as discussed in *Day One: Configuring JUNOS Interfaces and Routing* found at www.juniper.net/dayone.

Reaching a Domain Name System (DNS) Server

JUNOS can resolve hostnames to IP addresses if it knows the location of your DNS server(s). The approach is similar to the way Web browsers resolve the names of a Web site to its network address.

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

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

How to configure the DNS server:

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

```
set system name-server 172.26.27.2
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. JUNOS then uses this configured domain name as the default domain name to append to hostnames that are not fully qualified:

```
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. JUNOS then uses this list to set an order in which it appends domain names when searching for the IP address of a host.

```
set system domain-search [enterprise.com department.enterprise.com]
```

This command example configures JUNOS to search the enterprise.com and then the department.enterprise.com domains when attempting to resolve unqualified hosts.

VERIFY If you have configured your DNS server with the hostname and an IP address for your JUNOS 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 JUNOS 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 JUNOS device, you can either configure it manually, or your device can take a system time from an NTP (Network Time Protocol) server.

How to set time locally:

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

```
root@juniper1> set date 200901011200.00
```

The date is in the form (YYYYMMDDhhmm.ss).

How to use a remote time server:

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

Use the following steps to configure your device to use one or more NTP servers.

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

```
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 device can be up for a long time, and therefore the clock can drift:

```
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 JUNOS 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:

```
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 JUNOS and need to synchronize time with a remote time source, you can 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.

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

You can also check the NTP server status and associations of the clocking sources used by your device with the two following 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 JUNOS devices include physical interfaces for moving traffic in and out of the device, as well as special interfaces such as the management and loopback interfaces (see the prededing *Management Interface* and *Loopback Interface* sections).

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? *Day One: Configuring JUNOS Interfaces and Routing* discusses how to configure other types of interfaces. Check availability and download your copy at www.juniper.net/dayone.

Physical Interface Naming

Your platform may include Ethernet interfaces for carrying traffic and one or more of many wide area network interfaces available on different types of JUNOS devices. Regardless of the type of interface, JUNOS follows a standard format in its interface naming. The interface name is made up of two parts: the interface type and the interface numbering.

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

The Juniper engineers assign interface numbers corresponding to each interface location (in each hardware platform). Generally, they use the following conventions in designating numbers to the device interfaces, sequentially assigning numbers beginning with 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.

Putting it all together, an example of an interface name is `ge-0/0/1`. In this example, 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: Viewing the Hardware Configuration

Try entering the following operational mode command on your 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). JUNOS refers to these logical interfaces as *units*. Typically JUNOS requires that you set up one (or more) logical units on each physical interface.

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

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

Gigabit Ethernet

The best way to learn how to configure interfaces in JUNOS 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:

```
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 booklet, all of our configurations use `inet` which is how JUNOS 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.

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. Then, you can compare your listing to the configuration defined by the commands designated by an arrow ▶. The full configuration is available in the Appendix of the PDF version of this Day One booklet.

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 in 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 active ▶configuration as a series of set commands:

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 easier for you to see which commands created the configuration:

```
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 JUNOS displays the `show configuration | display set` listing from the top of the configuration mode, i.e. the `set` commands are in the form that 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 3

Setting Up User Accounts

JUNOS offers a rich and flexible set of features for configuring and managing user accounts, authentication, and permissions. This chapter gives you what you need on day one, as well as a few references for when you are ready to take advantage of the more advanced capabilities of JUNOS Software.

Creating Login Banners

You can create login banners for those who post messages and announcements to those who access the device. You might want to configure an initial login message now, before you create any user accounts.

Login Message

A login message displays 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` (newline, equivalent to a carriage return and line feed) as a delimiter:

```
set system login message “ Welcome \n to \n JUNOS Training\n “
```

After you set up the message above, any user accessing your device sees it displayed on their screen. For example, if the remote client is using the Secure Shell (SSH):

```
$ ssh juniper1
 Welcome
 to
 JUNOS Training
root@juniper1’s password:
```

TIP Use the login message to warn others that unauthorized access to your device is prohibited (ask your legal department for the preferred statement in your organization):

```
set system login message “WARNING: Unauthorized access is an
offense”
```

Login Announcement

Sometimes you want to make announcements only to authorized users *after* they have logged in. For example, you may want to announce an upcoming maintenance event. Use the `set system login announcement` command when you want to restrict your announcement to only authorized users:

```
set system login announcement “Maintenance scheduled 11PM to 2AM
tonight”
```

Configuring Login Accounts

JUNOS 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, as well as with local users and user templates that depend upon remote servers to provide authentication either using the RADIUS or TACACS+ protocols (discussed later).

Local User and Password

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

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, JUNOS assigns one for you. The format of the UID is a number in the range of 100 to 64000. 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 2, you use the `set` command to create a password as plain text, and JUNOS internally encrypts it:

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

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

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 *Systems Basics Guide* available at www.juniper.net/techpubs.

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, JUNOS checks the login class permission level for each command before accepting it. JUNOS comes with four pre-defined user login classes:

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

For the new user created in the example above, 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 more detailed permissions than are provided by 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. For example, you may want to create a custom login class just for network operations staff that you call *`netops`*:

```
set system login class netops
```

2. For each login class you can specify which permissions you want to allow or deny, but this example keeps things simple by giving the ***netops*** class access to everything:

```
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 chapter Configuring User Access in the *System Basics Configuration Guide* at www.juniper.net/techpubs.

Setting Up Remote Authentication

It's common practice to use remote authentication servers to centrally store information about users (see Chapter 1). You can configure JUNOS to use one or more remote authentication servers, including RADIUS and TACACS+ servers.

To set up remote authentication in your JUNOS 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 the method of using the `remote` template account. The username `remote` is a special case in JUNOS. 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 method, JUNOS 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 Creating a Local Template Account in the *Administration Guide* of JUNOS documentation at www.juniper.net/techpubs.

How to start the configuration for authentication by a RADIUS server:

Use the following steps to start the set up of user authentication by a RADIUS server. Complete the set up with steps 3 and 4 below.

1. Enter the RADIUS configuration statement:

```
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:

```
set system radius-server 172.26.27.5 port 1845
set system radius-server 172.26.27.5 secret Jun1p3r
```

How to start the configuration for authentication by a TACACS+ server:

Use the following steps to start the set up of user authentication by a TACACS+ server. Complete the set up with steps 3 and 4 below.

1. Enter the TACACS+ configuration statement:

```
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:

```
set system tacplus-server 172.26.27.6 port 49
set system tacplus-server 172.26.27.6 secret Jun1p3r
```

How to specify the login methods:

3. Specify the order in which JUNOS should attempt authentication:

```
set system authentication-order [ radius tacplus password ]
```

The above example assumes your network includes both RADIUS and TACACS+ Servers. Nonetheless, youmust include this command as part of the steps toconfigure the RADIUS and/or TACACS+ configurations. In this example, whenever a user attempts to log in, JUNOS begins by querying the RADIUS server for authentication. If it fails, it next attempts authentication with the TACACS+ server, and finally the locally configured user accounts.

ALERT! If the password option is not set (and the authentication server(s) are available), JUNOS does not make an attempt to authenticate with local passwords.

How to complete the configuration for authentication by a remote server:

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:

```
▶ set system login user adminjlk class super-user
▶ set system login user remote class super-user
```

How to verify the configuration of remote authentication server(s):

1. 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 5:

```
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
```

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
```

2. You can also 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
```

Enabling Remote Access

SSH, telnet, and FTP are widely-used standards for remotely logging into network devices and exchanging files between systems. 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 JUNOS.

SSH is telnet's successor and is the recommended method for remote access. SSH encrypts all traffic, including passwords, to effectively eliminate eavesdropping, connection hijacking, and other attacks. The SSH utility includes SCP (secure copy), 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:

```
  set system services ftp
  set system services telnet
▶ 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.

Committing Your Changes

Before leaving this chapter, don't forget to commit your work so that it becomes a part of the active configuration.

Chapter 4

Configuring SNMP

At this point, configuration of the base system and users is completed, so the next two chapters move on to basic configuration of the management functions in your device. Chapter 4 discusses how to configure Simple Network Management Protocol (SNMP), and Chapter 5 introduces additional management tools. The Day One goal is to integrate your new JUNOS device into your existing network management infrastructure as quickly and easily as possible.

Because many network management systems are currently based on SNMP, support for that protocol is a key feature of any network device's instrumentation.

JUNOS includes an onboard SNMP agent to provide remote management applications with access to a variety of detailed information. The SNMP agent in JUNOS supports SNMPv1, SNMPv2c, and SNMPv3 protocols, thereby enabling interoperability with a majority of the management applications on the market today. A set of both industry-standard and enterprise-specific management information bases (MIBs) are available.

ALERT! By default, JUNOS ships with the SNMP agent disabled. Follow the steps of this chapter to set up SNMP in your device.

Configuring SNMP Communities

Configuring the SNMP agent in JUNOS is a straightforward task that shares many familiar settings common to other managed devices in your network.

For example, you need to configure JUNOS with an SNMP community string and a destination for traps. Community strings are administrative names that group collections of devices (and the agents that are running on them) together into common management domains (see Chapter 1). Basically, if a manager and an agent share the same community, they can talk to one another.

An SNMP community defines the level of authorization granted to its members, such as which MIB objects are available, which operations (read-only or read-write) are valid for those objects, and which SNMP clients are authorized, based on their source IP addresses.

Try It Yourself: Getting Online Help

Before you begin, look at the online help available in operational mode as it relates to SNMP. Find out what configuration options JUNOS supports in your software version:

```
jadmin@juniper1> help reference snmp community
```

Get more detailed usage guidelines for a topic or configuration statement:

```
jadmin@juniper1> help topic snmp community
```

Finally, get a list of all the JUNOS Software commands where SNMP statements occur:

```
jadmin@juniper1> help apropos snmp
```

Configuration Groups

Make things easier for yourself by placing your common SNMP configuration settings inside a JUNOS configuration group (see Chapter 6). Not only does a configuration group help to keep all your customized configuration statements in one place, it also helps you to copy templates between devices later on.

To create a new configuration group that you name *common*, enter the following in configuration mode:

```
▶ edit groups common
```

You can now add your SNMP configuration statements to the *common* group and later apply them where you want in the configuration.

NOTE The examples in Chapters 4, 5, and 6 use *jadmin* as the user account.

How to create a read-only SNMP community:

1. Refer to what you recorded in Chapter 1 as the community string of the SNMP community used in your network. This example uses the de facto standard name *public* to create a community that gives limited read-only access:

```
[edit groups common]
▶ jadmin@juniper1# set snmp community public
```

2. Now focus your configuration statements onto this new branch of the configuration hierarchy:

```
edit snmp community public
```

3. Define the authorization level for the community:

```
[edit groups common snmp community public]
set authorization read-only
```

In the above command you are confining the `public` community to read-only access. Now any SNMP client (for example, an SNMP management system) that belongs to the *`public`* community can read MIB variables but cannot set (change) them.

4. Define a list of clients in the *`public`* community who are authorized to communicate with the SNMP agent in JUNOS. List the clients by IP address and prefix. Typically the list includes the SNMP network management system in your network, or the address of your management network. The following statement defines the network 192.168.1.0 (and any host within it) as being authorized:

```
set clients 192.168.1.0/24
```

5. Define the clients that are not authorized within the *`public`* community by specifying their IP address, followed by the `restrict` statement:

```
set clients 0.0.0.0/0 restrict
```

How to create a read-write SNMP community:

You can also define a community whose members have *read-write* access to the JUNOS device.

1. If you are following along from the previous example, go back up one level in the configuration hierarchy in order to create a new community. You also want to place this new community inside the JUNOS configuration group called *`common.`* Check that you are in this branch of the configuration by the edit banner:

```
jadmin@juniper1# up

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

2. This example uses the de facto standard community string *`private`* to identify the community granted read-write access to the SNMP agent running on the device:

```
edit community private
```

3. Use the following commands to set the authorization level, set the clients, and to specify the IP address for those with authorized access:

```
set authorization read-write
set clients 192.168.1.15/24
set clients 0.0.0.0/0 restrict
```

NOTE A later section discusses how to control access to specific branches of the SNMP MIB tree by defining MIB views and assigning them to SNMP communities.

Configuring SNMP Traps

Traps are unsolicited messages sent from an SNMP agent to remote network management systems or trap receivers. Many enterprises use SNMP traps as part of a fault-monitoring solution, in addition to system logging (see Chapter 5). In JUNOS, SNMP traps are not forwarded by default, so you must configure a trap-group if you wish to use SNMP traps.

How to configure a trap-group:

1. Create a single, consistent source address that JUNOS applies to all outgoing traps in your device. A source address is useful, because although most JUNOS devices have a number of outbound interfaces, using one source address helps a remote NMS to correlate the source of the traps to an individual device:

```
jadmin@juniper1# edit groups common snmp

[edit groups common snmp]
jadmin@juniper1# set trap-options source-address lo0
```

The above commands place the statement inside the *common* configuration group that you have defined. It uses the IP address of the loopback interface `lo0` (see Chapter 2) as the source address for all the SNMP traps that originate from the device.

2. Create a *trap-group* where you can list the types of traps to be forwarded and the targets (i.e., addresses) of the receiving remote management systems:

```
set trap-group managers version v2 targets 192.168.1.15
```

The above command creates a trap-group called *managers* which allows SNMP version 2 formatted notifications (i.e., traps) to be sent to the host at address 192.168.1.15. This statement forwards all categories of traps.

3. Use the `categories` statement to define the specific subset of trap categories to be forwarded.

```
▶ set trap-group managers version v2 targets 192.168.1.15
categories authentication
```

Table 4.1 lists a variety of trap categories used in JUNOS.

Table 4.1 **JUNOS SNMP Trap Categories**

Configuration Option	MIB	Description
`authentication`	Standard MIB-II	Authentication failures on the agent (the device)
`chassis`	Juniper proprietary	Chassis and router environment notifications
`configuration`	Juniper proprietary	Configuration mode notifications
`link`	Juniper proprietary	Interface transitions, such as transitioning from up to down
`rmon-alarm`	Juniper proprietary	SNMP Remote monitoring events
`routing`	Juniper proprietary	Routing protocol notifications
`startup`	Standard MIB-II	Router reboots (soft/warm and full reboots)

Applying the Configuration Group

So far in this chapter you have been placing all of your SNMP configuration statements inside a JUNOS configuration group named *common*. And you've used a configuration group with the intention of making things easier later on. At some point in the future you might want to reuse the *common* statements elsewhere in the configuration, or copy them to other devices.

Now, in order for the statements within the *common* configuration group to be recognized by the JUNOS Software, you have to apply them:

```
jadmin@juniper1# top

[edit]
jadmin@juniper1# set apply-groups common
```

This example applies the JUNOS configuration group called *common* at the top of the configuration, so the group applies to all parts of the configuration.

ALERT! Where you apply configuration groups within the configuration is important, as only that specific hierarchy level (and below) inherits the group statements. Further, the ordering of configuration groups is important, as JUNOS inherits statements in the order that they are applied.

VERIFY Have you committed the trap settings? If so, you can perform a quick test to confirm that you have configured SNMP traps correctly. This example generates an authentication failure trap (i.e., the SNMP agent received a request with an unknown community), although other traps types can also be spoofed:

```
jadmin@juniper1> request snmp spoof-trap authenticationFailure
Spoof-trap request result: trap sent successfully
```

Configuring SNMP System Details

You can use SNMP to store basic administrative details, such as a contact name and the location of the device. Your management system can then retrieve these remotely, when you're troubleshooting a problem or performing an audit. In SNMP terminology, these are the *sysContact*, *sysDescription*, and *sysLocation* objects found within the system group of MIB-2 (as per RFC 1213). You can set initial values directly via a simple JUNOS configuration process.

How to set the system contact details:

1. Set the system contact details by including the `contact` statement in the `[edit snmp]` hierarchy of the configuration, or under an appropriate configuration group as explained above:

```
set contact “For help, please email support@enterprise.com”
```

2. Set the system description:

```
▶ set description “Juniper EX4200”
```

3. Set the system location:

```
▶ set location “London Corporate Office”
```

VERIFY After you have committed the configuration with these changes, you can perform a quick test to confirm that you have entered the system details correctly. Enter the following operational mode command:

```
jadmin@juniper1> show snmp mib walk system
sysDescr.0    = Juniper EX4200
sysObjectID.0 = jnxProductNameEX4200
sysUpTime.0   = 85957438
sysContact.0  = For help, please email support@enterprise.com
sysName.0     = junos
sysLocation.0 = London Corporate Office
sysServices.0 = 4
```

The `show snmp mib walk system` command performs a MIB walk-through of the system table (from MIB-2 as per RFC 1213). The SNMP agent in JUNOS responds by printing out each row in the table and its associated value. You can use the same command to perform a MIB walk-through of any part of the MIB tree supported by the agent.

Configuring View Based Access Control

SNMPv3 defines the concept of *MIB views* (so-called “View Based Access Control,” see RFC 3415), which gives an agent better control over who can access specific branches and objects within its MIB tree. A *view* simply consists of a name and a collection of SNMP object identifiers, which are either explicitly included or excluded. Once defined, a view is then assigned to an SNMPv3 group or SNMPv1/v2c community (or multiple communities), automatically masking which parts of the agent's MIB tree members of the group or community can (or cannot) have access to.

How to create a MIB view:

Although most network management systems have recently started using SNMPv3, a feature of the SNMP agent in JUNOS is that it allows use of MIB views with both SNMPv1 and SNMPv2c communities. This example shows you how.

1. Enter the configuration mode of the JUNOS CLI and include the `view` statement to create *`ping-mib-view`*:

```
set snmp view ping-mib-view oid 1.3.6.1.2.1.80 include
```

NOTE The `oid` statement does not require a dot at the beginning of the object identifier.

The `snmp view` statement now includes the branch underneath the object identifier .1.3.6.1.2.1.80 (i.e. the whole of the DISMAN-PING-MIB subtree, as per RFC 2925), which effectively permits access to any object under that branch.

2. Add a second branch in the same MIB view:

```
set snmp view ping-mib-view oid jnxPingMIB include
```

NOTE In this example, the object identifier is referred to by its object identity name, rather than by its dotted oid number.

The additional `snmp view` statement extends the MIB view created above, to include the Juniper Networks enterprise specific extensions to the DISMAN-PING-MIB, which are located in the jnxPingMIB branch (under .1.3.6.1.4.1.2636.3.7).

How to assign a MIB View to a community:

With the MIB view created, it is a simple task to assign it to the appropriate community that you want to control. This example creates a new community *`ping-mib`* which has read-write access to create entries within the DISMAN-PING-MIB.

1. Give read-write access to the community as a whole:

```
set snmp community ping-mib authorization read-write
```

2. Associate the MIB view created earlier with the new community:

```
set snmp community ping-mib view ping-mib-view
```

Once you have committed the configuration, any member of the `ping-mib` community now has read/write access to the branches that you specified under `ping-mib-view`.

Committing Your Changes

Commit your work and then you can go to the supplemental Appendix included in the PDF version of this booklet to compare your configuration to the provided example.

Chapter 5

Using Monitoring and Logging

In addition to SNMP, JUNOS offers other management tools, including the JUNOS health monitor, remote monitoring, system logs, and J-Web for web based management. Configuring these additional tools to monitor and control your device completes your set up of the JUNOS basics.

Discovering the JUNOS Health Monitor

Enterprise networks may include so many networked devices that it becomes impractical to use a central network management system to poll all of them. A more scalable approach is to rely on the network devices themselves to notify the NMS when something requires attention. JUNOS provides a useful mechanism called the health monitor, allowing a device to monitor its own key operating metrics and trigger alarms when normal operating parameters are exceeded.

The JUNOS health monitor uses the underlying remote monitoring (RMON) mechanism (described in the next section) to monitor a selection of predefined object instances, and to extend the basic RMON functionality to support dynamic object instances. Examples of predefined object instances are file system usage, CPU usage, and memory usage. Dynamic object instances include software processes and other entities that are only known at run time. Unlike standard RMON, the JUNOS health monitor does not require device-specific expertise in order to create a useful monitoring application. Once you configure the health monitor, you can forget about it until it triggers an event.

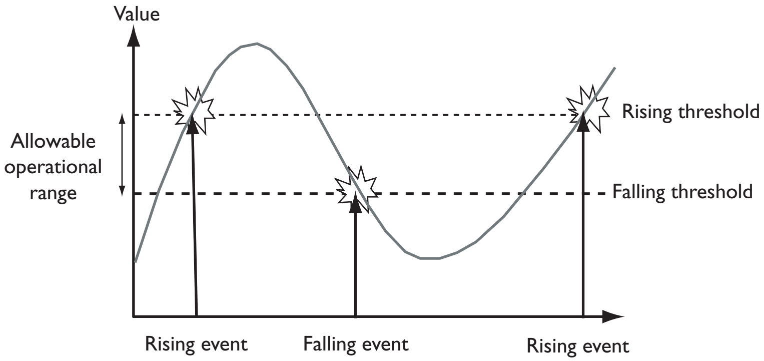
Both RMON and the JUNOS health monitor use the concept of rising and falling thresholds to define an allowable operating range for an object, as shown in Figure 5.1 on the next page. Each time a monitored object crosses a threshold in Figure 5.1, it triggers an event. A rising event raises an alarm and a falling event clears (or resets) an alarm. This provides useful information that a remote NMS can act upon.

How to configure the JUNOS health monitor:

1. Add the `health-monitor` statement to the SNMP hierarchy of the configuration:

```
[edit snmp]
▶ set health-monitor
```

Figure 5.1 **RMON Thresholds**



The configuration example uses the health-monitor defaults, setting the rising threshold to 80%, the falling threshold to default to 70%, and the polling interval to default to 300 seconds (5 minutes).

2. You can change the threshold values from the defaults. Set the rising and falling threshold percentages to lower values and adjust the frequency (in seconds) used by health monitor to check monitored objects:

```
[edit snmp health-monitor]
set rising-threshold 70
set falling-threshold 60
set interval 900
```

NOTE If you have enabled SNMP traps, health monitor generates RMON rising and falling threshold events, and if you have enabled syslog, health monitor uses the HEALTHMONITOR syslog tag.

Monitoring Devices Remotely

Many enterprises use the RMON capabilities of SNMP agents to monitor aspects of their IP network. For example, your network may use dedicated devices called *RMON probes* that analyze traffic for statistical reporting. Or, perhaps it relies upon embedded agents on devices such as IP routers and Ethernet switches. JUNOS provides an embedded RMON agent capability, and supports the RMON alarm and event groups from the Remote Network Monitoring MIB defined in RFC 2819.

RMON alarms and events are complementary to each other. You configure alarms to tell the RMON agent to periodically take statistical samples from a list of variables and compare them to thresholds. If a monitored variable crosses a threshold, JUNOS generates a corresponding event, which sends a notification (SNMP trap or syslog message) out from the device.

This section describes how you can configure RMON alarms and events via the JUNOS CLI. Since the MIB groups are writable, if write access is configured, an SNMP management system can also remotely configure both alarms and events (see the *View Based Access Control* section in Chapter 4).

For example, assume that you want to monitor the number of times that remote hosts try to query the SNMP agent in JUNOS using an incorrect community – which could indicate that someone is attempting something malicious, or that an NMS has been misconfigured. The value is reported via SNMP with the `snmpInBadCommunityNames` object (see the SNMPv2 MIB as per RFC 1907). Because it is an integer value, it can easily be monitored by the RMON agent in JUNOS.

How to create an RMON alarm:

When you create an RMON alarm, you can also define alarm parameters, such as the object that you want to monitor, the monitoring interval, and any thresholds, as well as the associated event.

1. Create an RMON alarm under the `[edit snmp]` configuration branch:

```
[edit snmp]
set rmon alarm 100
```

2. Adjust the parameters of the RMON alarm:

```
[edit snmp rmon alarm 100]
set variable snmpInBadCommunityNames.0
set sample-type delta-value
set rising-threshold 100
set falling-threshold 90
set interval 900
set falling-event-index 100
set rising-event-index 100
```

In this example `snmpInBadCommunityNames` is a counter object, which is incremented – every time the JUNOS device receives an SNMP request with or for an unknown community. If the device receives more than

100 such requests in the last 900 seconds (15 minutes), the alarm triggers event number 100 to run.

3. Now create the event associated with your alarm:

```
[edit snmp]
set rmon event 100

[edit snmp rmon event 100]
set community managers
set description "snmpInBadCommunityName threshold event"
set type log-and-trap
```

This event generates an SNMP trap whenever it is triggered, sending it to the trap group `managers`. It also creates a new entry in the RMON logTable (for details see RFC 2819). These entries can be useful if you use an NMS to independently collect statistics about the number of RMON events generated by the device over time.

VERIFY After committing your changes, you can perform a quick test to check that you have entered the RMON alarm and event entries correctly. Enter the following command from operational mode:

```
jadmin@juniper1> show snmp mib walk rmon
```

The `show snmp mib walk rmon` command performs a MIB walk through the rmon table (per RFC 2819), printing out each row and its associated value. You can use this same command to perform a MIB walk through any part of the MIB tree supported by the JUNOS agent.

TIP You can troubleshoot the RMON operations by querying the contents of the Juniper Networks enterprise RMON MIB, jnxRmon. The Juniper MIB provides valuable information on alarm entries that fail to trigger which, for example, might occur if an OID that was specified in an alarm entry did not exist.

MORE? JUNOS can do much more than just generate an SNMP trap when an event is triggered. For example, it can correlate multiple matching events to remove duplicates, it can send a syslog message (see below), and it can even perform a configuration change. If you want JUNOS to perform more actions when an event is triggered, refer to JUNOS Event Policy in the *Configuration and Diagnostic Automation Guide* available at www.juniper.net/techpubs.

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. 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. JUNOS 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 JUNOS syslog event messages to a central server for fault de-duplication and analysis.

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

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

Syslog Destinations

JUNOS Software offers the flexibility to send syslog messages to a variety of different destinations, as listed in Table 5.1.

Table 5.1 **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 booklet.

Message Facilities and Levels

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

JUNOS supports a number of syslog facilities to specify the software process that generates each message. If you are unsure of which facility you want to monitor, or 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 varying level, i.e., 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 JUNOS is shown in Table 5.2.

Table 5.2 **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

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 JUNOS Software logs messages to that level *and above*, notifying you of any error, critical, alert, and emergency level messages.

MORE? For general information about configuring syslog, see the technical documentation manual *System Basics Configuration Guide*. For a full list of all of the messages supported by JUNOS, refer to the *System Log Messages Reference*. Both are available at www.juniper.net/techpubs/.

Sending Messages to a File

Now that you are familiar with syslogs and what they do in JUNOS, the simplest and the most useful syslog configuration on day one 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:

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. JUNOS 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 JUNOS 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
```

Here 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. JUNOS 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:

To forward messages to a user use these configuration steps.

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

```
set system syslog user * any emergency
set system syslog user jadmin any critical
```

The first `set` command forwards any emergency log message to all users, as indicated by the JUNOS 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 located remotely.

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:

```
set system syslog console any error
```

Forwarding Messages to a Remote Server

JUNOS 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:

```
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

Many enterprises use commercial fault monitoring software. 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 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

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 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*:

```
▶ set system syslog host loghost log-prefix JUNOS
```

This 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 coming from the device.

Including Priority Information

An event's facility and severity level are together referred to as its priority. However, while the priority is known to 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:

```
▶ set explicit-priority
```

The `explicit-priority` statement inserts the priority in the form FACILITY-LEVEL into the beginning of each message. Note that in this case, the level is a numerical value as previously listed in Table 5-2.

Facility Override

Some third party syslog monitoring applications “listen” to messages arriving with a specific facility. Since JUNOS 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. You must also configure the third party syslog application to handle the messages in the desired manner.

MORE? Syslog is one of several mechanisms that log network events in JUNOS. The other commonly used tool is trace logging (also known as traceoptions). Trace logging keeps track of specific processes, such as routing packets sent and received. Traceoptions output is similar to debug output in other systems. To learn about trace logging, refer to the *System Basics Configuration Guide* available at www.juniper.net/techpubs.

Using Web-Based Management

Many people prefer to use a web browser as the main application for accessing their network devices to monitor, administer, and troubleshoot them. JUNOS features a rich web-based interface called *J-Web* for you to use.

This section walks you through the basic steps that you need to know on day one: how to check if J-Web is installed, how to install it, and how to configure it.

Installing J-Web

J-Web is a commercial product that comes preinstalled on EX Series Ethernet switches, J Series routers, and SRX Series services gateways. On other platforms, such as the M Series and MX Series routers, you need to obtain and install the software package to use it. This section describes how you can obtain J-Web and install it.

How to check if J-Web is installed:

To find out if J-Web is preinstalled on your device and its version:

1. Run the following operational mode command:

```
jadmin@juniper1> show version detail | match Web
```

TIP The `match` statement supports regular expressions, making it a very flexible way for you to reduce clutter in output.

If J-Web is installed, show version will report "JUNOS Web Management" with a version number, and you can skip the rest of this section and move to *Configuring J-Web*.

2. If J-Web is not installed, you have to obtain it and then install it. It is important to determine which version of JUNOS you are running and install the version of J-Web that matches exactly. The show version command tells you what you need to know:

```
jadmin@juniper1> show version
Hostname: junos
Model: m10
JUNOS Software Release [9.4R2.9]
```

How to install the J-Web package:

Obtain the J-Web software package either from your reseller or directly from the Juniper Networks support website www.juniper.net/customers/csc/software under the JUNOS Software download section.

JUNOS can also pull the package from a remote server if you specify the location of the package as a URL with the `request system software` command. The following example uses FTP as the protocol to pull the package onto the JUNOS device:

```
jadmin@juniper1> request system software add ftp://ftp:secret@server/pub/junos/jweb-
9.4R2.9-signed.tgz
Installing package 'ftp://ftp:secret@server/pb/junos/jweb-9.4R2.9-signed.tgz' ...
Verified jweb-9.4R2.9.tgz signed by PackageProduction_9_4_0
Adding jweb...
Available space: 671134 require: 8226
Mounted jweb package on /dev/md9...
Verified manifest signed by PackageProduction_9_4_0
Executing /packages/mnt/jweb-9.4R2.9/mount.post..
Reloading /config/juniper.conf.gz ...
Activating /config/juniper.conf.gz ...
mgd: commit complete
Restarting mgd ...
Saving package file in /var/sw/pkg/jweb-9.4R2.9-signed.tgz ...
Saving state for rollback ...

WARNING: cli has been replaced by an updated version:
CLI release 9.4R2.9 built by builder on 2009-03-25 07:29:27 UTC
Restart cli using the new version ? [yes,no] (yes)

Restarting cli ...
jadmin@juniper1>
```

Configuring J-Web

With the J-Web package installed on your JUNOS device, you can go ahead and configure it. Since J-Web is not configured and running by default, you must complete its initial setup via the JUNOS CLI. This is true for all JUNOS devices:

```
set system services web-management http
```

Once you have committed the configuration change you can access the JUNOS Software via the J-Web interface. Simply enter the hostname or IP address of the JUNOS device as a URL into a web browser.

For further details on how to use the J-Web interface, refer to the *J-Web Interface User Guide*, available online at www.juniper.net/techpubs/. J-Web has differences for different devices, so be sure to get the guide for your specific platform.

Reviewing Your Work

This chapter concludes the basic setup of your device. Chapter 6 provides you with additional commands and shortcuts that can ease configuration moves, adds, and changes in your device.

If you followed along in your own device by entering the commands designated by an arrow ▶, your configuration should now match the configuration listing provided in the Appendix of the PDF version of this Day One booklet. Once again, the PDF is freely available at www.juniper.net/dayone.

Chapter 6

Working with Configuration Templates and Other Shortcuts

This chapter provides some handy techniques that can save you a lot of time when you are creating and modifying configurations in the command-line interface. These techniques let you easily reuse configuration statements set up in 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 can not only speed configuration editing, but can also help reduce errors often associated with repetitive command entry.

Several of the examples go beyond what you would typically do on day one, but provide you with a preview to other powerful features within JUNOS.

TIP In this booklet, the headings in the left margin are generally named after the commands themselves, for easier look up.

Displaying Set Commands

One of the most frequently used and easiest configuration shortcuts is displaying an existing set of commands known to work and then reusing them somewhere else:

```
jadmin@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
```

Regardless of where you specifically entered your `set` commands in the configuration hierarchy, the `show configuration | display set` command lists them as if they were entered from the top of configuration mode. You can use the command to display the whole configuration, or just a portion as shown in the above example.

In the displayed listing, look for the `set` command that you want to re-use, and copy it using an available copy command such as the keyboard command Control+C. Then, move your cursor to where you want to re-use the command, and paste it (you can use the keyboard command Control+V). You can cut and paste the entire output (the returns are embedded at the end of each line). Before you hit Enter you can make any other changes needed in the command lines, such as changing the IP address.

SHORTCUT You can combine the top command with other commands to enter new statements at the top of the hierarchy, regardless of where you are currently working in the configuration. Example:

```
top set interfaces ge-0/0/1 unit 0 family inet address
192.168.100.1/30
```

Making Changes Faster

Networks are complex, dynamic systems that may need to change frequently in order to meet new business needs. Fortunately JUNOS offers helpful commands to shorten the time that it takes to make changes in the existing configuration.

Rename

Sometimes you may want to rename a section of the JUNOS configuration, for example, to alter an older naming convention to adhere to a new policy. Let's assume you want to rename ge-0/0/0 to the new naming convention of ge-1/0/0. Below is the current configuration snippet:

```
jadmin@juniper1# show interfaces ge-0/0/0
unit 0 {
    family inet {
        address 100.100.100.1/24;
    }
}
```

You can rename the interface in one step with the rename command:

```
jadmin@juniper1# rename interfaces ge-0/0/0 to ge-1/0/0
```

VERIFY The show command lets you check that the change has occurred:

```
jadmin@juniper1# show interfaces ge-1/0/0
unit 0 {
    family inet {
        address 100.100.100.1/24;
    }
}
```

NOTE In this context, the JUNOS rename command acts in a similar way to the Unix "mv" (move) command by renaming the original section rather than making a copy of it.

Copy

JUNOS also allows you to make copies of parts of the configuration by using the copy command. For example, let's assume that you created a template for a local user called *`logintemplate`*, and now you want to make a copy for a new user "joe" who has recently joined your team:

```
jadmin@juniper1# show system login user
user logintemplate {
   full-name "Generate network operations user";
   class netops;
   authentication {
      encrypted-password "$1$Naeta3Iw$./sgTTPK0NoH0PJdsXvP6.";
## SECRET-DATA
   }
}
```

You can make a copy of this template for a new user "joe" by using the copy command:

```
jadmin@juniper1# edit system login
jadmin@juniper1# copy user logintemplate to user joe
```

VERIFY Check that JUNOS has created the new local user:

```
jadmin@juniper1# show
user logintemplate {
   full-name "Generate network operations user";
   class netops;
   authentication {
      encrypted-password "$1$Naeta3Iw$./sgTTPK0NoH0PJdsXvP6.";
## SECRET-DATA
   }
}
user joe {
   full-name "network operations user";
   class netops;
   authentication {
      encrypted-password "$1$Naeta3Iw$./sgTTPK0NoH0PJdsXvP6.";
## SECRET-DATA
   }
}
```

To complete the setup of `joe`, you can modify his password, and he's ready to go.

Replace

Another useful command is `replace`, which changes a given character string throughout the configuration to something else. For example, assume that you have interface `ge-0/0/0` referenced in the protocols branch of the configuration because you have configured OSPF on it, as shown below:

```
jadmin@juniper1# show interfaces ge-0/0/0
unit 0 {
   family inet {
      address 100.100.100.1/24;
   }
}
jadmin@juniper1# show protocols ospf
area 0.0.0.0 {
   interface ge-0/0/0.0;
}
```

In this example, you can use the `replace` command to rename the interface to the new naming convention throughout the entire configuration:

```
jadmin@juniper1# replace pattern ge-0/0/0 with ge-1/0/0
```

Insert

The `insert` command allows you to insert a configuration statement either before or after an item in an ordered sequence. It is especially useful if you are configuring firewall filters and routing policies and need to change the ordering of terms. For instance, suppose you have the following policy:

```
[edit policy-options policy-statement multiterm]
jadmin@juniper1# show
term reject {
   then reject;
}
term accept {
   from protocol bgp;
   then accept;
}
```

Applying this to a BGP import or export policy would have it reject all routes, because the terms are processed from the top down, and the reject term is before the accept term. To adjust, use the `insert` command:

```
[edit policy-options policy-statement multiterm]
jadmin@juniper1# insert term accept before term reject
```

You can insert `before` or `after` depending on the order needed for the policy.

MORE? Learn more about configuring policies with the *Policy Framework Configuration Guide* available at www.juniper.net/techpubs.

Defining Groups

For 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. Use them to create smaller, more logically constructed configuration files. Not only is 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 booklet has previously shown how to set up a group for SNMP management in Chapter 4. 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 important, as JUNOS 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 SONET 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:

These commands set up a group called *sdh* and apply it throughout the configuration.

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-*>`. The `<so-*>` 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.

```
set apply-groups sdh
```

ALERT! If you are using JUNOS configuration groups, by default 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;
   }
}
```

TIP 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:

The example in this section shows 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 thoughout:

```
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 an L3 VPN.

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

```
set interfaces ge-0/0/0 apply-groups-except isis-mpls
```

Using `apply-groups-except` tells JUNOS to exclude the *`isis-mpls`* group on the `ge-0/0/0` interface, even though the *`isis-mpls`* group has been applied at the top of the configuration.

MORE? For full details on applying configuration groups, refer to the section on Configuration Groups in the *CLI User Guide* at www.juniper.net/techpubs/.

Using Configuration Templates

Let's say you are working on a large network, where you are responsible for installing and configuring 100 new Juniper Networks devices (without the help of a management system), with only one 2-day weekend to finish the job. Because most of the devices operate in the same network, many of the configuration parameters are the same for all of them. Instead of typing configuration data individually into each device, it is 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 the candidate into a file, by providing 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'
```

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: /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 JUNOS to merge the current candidate configuration with the contents of the loaded file. JUNOS adds the template statements, exactly as you saved them, to the [edit groups common] hierarchy location of the device configuration.

TIP The above 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.

Saving Your Work

This section presents more examples of ways to save the candidate or the active configuration. It can be easy to forget which configuration files you are saving, so the words *candidate* and *active* are underlined for quick reference.

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

MORE? *Day One: Exploring the JUNOS CLI* introduced the operational mode file commands for uploading and downloading files from servers.

How to save a candidate file locally:

Every JUNOS user defined in the configuration has their own home directory within the device in the form: `/var/home/username`. 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’
```

VERIFY You can check your home directory on the device using the `file list` command from operational mode:

```
jadmin@juniper1# run file list
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, and reuse these command blocks in other devices in your network. For example, use the same system login information for all the switches in the network.

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

How to save a configuration file remotely:

The following 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’
```

TIP If you want to save the active configuration, you can use the operational mode `file copy` command (refer to *Day One: Exploring the JUNOS CLI*). 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 on an interval:

Let's say you are making a copy of your JUNOS archive every Tuesday by logging in and making the copy. Wouldn't it be easier if JUNOS could do this for you, not just on Tuesdays but every day? You can configure JUNOS to automatically save the latest active configuration file and transfer it to a remote host.

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

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

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

```
set system archival configuration transfer interval 1440
```

This configures JUNOS to take the active configuration and send a copy to the ftp `remot` server in the directory `archives` every 1440 seconds (every 24 hours, once per day).

How to automate saving the active configuration upon commit:

Another option is to configure JUNOS to archive the active configuration after every commit (i.e. every time the configuration has been changed):

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

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

2. Configure JUNOS so that it transfers the active configuration after every commit:

```
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 safekeeping.

Loading Configurations

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

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
```

ALERT! Remember that 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 a device. For example, 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. JUNOS adds these statements, as you saved them, to the `[edit system login]` hierarchy location of your configuration.

NOTE 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. The relative option discussed below lets you specify where JUNOS 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 JUNOS 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 such as Control+C. Then enter the load merge terminal command on the destination router, and paste the snippet in, for example, by using the 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 such as Control+C, making sure you include the end curly bracket.

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

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

Then issue the `load` command as before but 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 the above example: `load merge filename relative`. (Find additional examples of how you can use load commands in the *CLI User Guide* available at www.juniper.net/techpubs.)

What to Do Next & Where to Go ...

www.juniper.net/dayone

If you're reading a print version of this booklet, go here to download the PDF version which includes supplemental info in its Appendix. Also, find out what other Day One booklets are currently available.

www.juniper.net/junos

Everything you need for JUNOS adoption and education.

http://forums.juniper.net/jnet

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

www.juniper.net/techpubs

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

www.juniper.net/books

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

www.juniper.net/training/fasttrack

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

Appendix

Note: This Appendix only appears in the PDF edition of *Day One: Configuring JUNOS Basics*.

Configuration Information Worksheet

Hostname

Management Port IP Address

Management Port Network Prefix

Loopback Interface IP Address

Backup Router IP Address

Domain Name Server IP Address(es)

Network Time Protocol Server IP Address

You should keep all your passwords secret at all times. If you write down a password here, make sure that this document is kept safe and secure.

Initial Root Password

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

Day One: Configuring JUNOS Basics Configuration Listing

This section provides the configuration listing for all the statements that this booklet has helped you to configure on your device. The listing provided on the following pages is the resulting configuration if you entered all the commands that are designated by an arrow (in the left margin) exactly as they are shown. These commands are found in Chapters 2 through 5 of this booklet.

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: 2009-06-16 08:32:35 CEST by root
version “9.5I0 [builder]”;
groups {
    common {
        snmp {
            community public {
                authorization read-only;
                clients {
                    192.168.1.0/24;
                    0.0.0.0/0 restrict;
                }
            }
            community private {
                authorization read-write;
                clients {
                    192.168.1.15/24;
                    0.0.0.0/0 restrict;
                }
            }
            trap-options {
                source-address lo0;
            }
            trap-group managers {
                version v2;
                categories {
                    authentication;
                }
                targets {
                    192.168.1.15;
                }
            }
        }
    }
```

```
}
apply-groups common;
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 "$9$KyEWXNs2aikP4oT39Cu0LxN";
        }
    }
    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$D9KVQKofqwKMEfcBjp
3zg0”;
            }
        }

        user remote {
            uid 2001;
            class super-user;
        }
        user adminjlk {
            uid 2002;
            class super-user;
        }
    }
    services {
        ssh;
        web-management {
            http;
        }
    }
    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 {
        unit 0 {
            family inet {
                address 192.168.100.1/30;
            }
        }
    }
    me0 {
        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;
          }
       }
   }

}
snmp {
   description "Juniper EX4200";
   location "London Corporate Office";
   contact "For help, please email support@enterprise.com";
   view ping-mib-view {
      oid 1.3.6.1.2.1.80 include;
      oid jnxPingMIB include;
   }
   community ping-mib {
      view ping-mib-view;
      authorization read-write;
   }
   community managers;
   rmon;
   health-monitor {
      interval 900;
      rising-threshold 70;
      falling-threshold 60;
   }
}
```

Command Reference

(**Summary of commands from** ***Day One: Using the JUNOS CLI.***)

Configuration Mode Commands

activate Activate a portion of the configuration.

annotate Leave comments about a configuration statement.

commit Commit candidate set of changes.

commit at Commit candidate at a set time.

commit check Validate the candidate configuration without activating any changes.

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

compare Display the differences between the two configurations.

copy Copy a statement.

deactivate Mark portions of the configuration as inactive.

delete Remove a configuration statement(s) or identifier.

edit Move to the designated hierarchy level.

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

exit configuration-mode Exit configuration mode.

help Get onboard help.

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

rename Assign a new name to a configuration or identifier.

rollback Restore the candidate file to a previous committed configuration.

run Run an operational mode command.

set Create a statement hierarchy and set identifier values.

show Display the candidate configuration.

top Move to the first hierarchy level.

up Move up one level in the hierarchy.

Operational Mode Commands

clear Remove system information.

configure Enter configuration mode.

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

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

exit Exit operational mode.

file copy Create and archive files.

file list Lists files and directories on the device.

file show View the file contents.

help Get onboard help.

monitor Show real-time debugging info.

ping Send a message to another host to verify connectivity.

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

request Install new software versions, reboot, shut down.

restart Restart individual operating system daemons.

set Establish system properties.

show Show system information.

ssh Start secure shell on another host.

start shell Log in to the C shell interface.

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

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

Command Reference

(Summary of commands from *this* Day One booklet.)

Configuration Mode Commands

commit Commit candidate set of changes.

commit synchronize Commit candidate set of changes to both routing engines of a single device.

commit comment Add a comment that describes the committed configuration.

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

commit and-quit Save software configuration changes, activate the configuration, and exit configuration mode.

compare Display the differences between the two configurations.

copy Copy a statement.

edit Move to the designated hierarchy level.

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

help Get onboard help.

insert Insert a new ordered data element

load merge Combine the current configuration and the loaded configuration.

load override Replace the current configuration with the loaded configuration.

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

rename Assign a new name to a configuration or identifier.

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

replace Replace character string in configuration.

rollback Go back to the previous committed configuration.

run Run an operational mode command.

save Save configuration to ASCII file.

set Create a statement hierarchy and set identifier values.

set apply-groups Apply a configuration group to a specific hierarchy level in a configuration.

set groups Create a configuration group.

set interfaces Interfaces on this device.

set snmp SNMP configurations on this device

set system Create a statement hierarchy and set identifier values in the system branch of the configuration.

> authentication-order Order in which authentication methods are invoked.

Operational Mode Commands

clear Remove system information.

configure Enter configuration mode.

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

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

exit Exit operational mode.

file copy Create and archive files.

file list Lists files and directories on the device.

file show View the file contents.

help Get onboard help.

monitor Show real-time debugging information.

ping Send a message to another host to verify connectivity.

Command Reference

(continued from previous page)

Configuration Mode Commands

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

request Install new software versions, reboot, shut down.

restart Restart individual operating system daemons.

set Establish system properties.

show Show system information.

ssh Start secure shell on another host.

start shell Log in to the C shell interface.

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

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