JUNOS™ Internet Software
for J-series™, M-series™, and T-series™
Routing Platforms

CLI User Guide

Release 7.6

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net

Part Number: 530-015682-01, Revision 1

JUNOS Internet Software for J-series, M-series, and T-series Routing Platforms CLI User Guide, Release 7.6
Writing: Lisa Kelly
Editing: Sonia Saruba
Illustration: Nathaniel Woodward
Cover design: Edmonds Design

Revision History
27 March 2006—Revision 1

The information in this document is current as of the date listed in the revision history.

Year 2000 Notice

Juniper Networks hardware and software products are Year 2000 compliant. The JUNOS software has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

Software License

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

End User License Agreement

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. The Parties. The parties to this Agreement are Juniper Networks, Inc. and its subsidiaries (collectively "Juniper"), and the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. The Software. In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, and updates and releases of such software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller. "Embedded Software" means Software which Juniper has embedded in the Juniper equipment.

3. License Grant. Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use the Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius software on multiple computers requires multiple licenses, regardless of whether such computers are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. Use Prohibitions. Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use the Embedded Software on non-Juniper equipment; (j) use the Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. Audit. Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. Confidentiality. The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. Ownership. Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. Warranty, Limitation of Liability, Disclaimer of Warranty. The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. Termination. Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. Taxes. All license fees for the Software are exclusive of taxes, withholdings, duties, or levies (collectively "Taxes"). Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software.

11. Export. Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. Commercial Computer Software. The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. Interface Information. To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. Third Party Software. Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at http://www.gnu.org/licenses/gpl.html, and a copy of the LGPL at http://www.gnu.org/licenses/lgpl.html.

15. Miscellaneous. This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

Table of Contents

Part 3 Advanced Features

Part 4 CLI Command Summaries

About This Guide

This preface provides the following guidelines for using the *JUNOS Internet Software for J-series, M-series, and T-series Routing Platforms CLI User Guide* and related Juniper Networks, Inc., technical documents:

Objectives

This guide describes how to use the JUNOS command-line interface (CLI) to configure, monitor, and manage Juniper Networks routing platforms.

NOTE: This guide documents Release 7.6 of the JUNOS Internet software. For additional information about the JUNOS software—either corrections to or information that might have been omitted from this guide—see the software release notes at http://www.juniper.net/.

Audience

This guide is designed for network administrators who are configuring and monitoring a Juniper Networks J-series, M-series, or T-series routing platform.

To use this guide, you need a broad understanding of networks in general, the Internet in particular, networking principles, and network configuration. You must also be familiar with one or more of the following Internet routing protocols:

- Border Gateway Protocol (BGP)
- Distance Vector Multicast Routing Protocol (DVMRP)
- Intermediate System-to-Intermediate System (IS-IS)
- Internet Control Message Protocol (ICMP) router discovery
- Internet Group Management Protocol (IGMP)
- Multiprotocol Label Switching (MPLS)
- Open Shortest Path First (OSPF)
- Protocol-Independent Multicast (PIM)
- Resource Reservation Protocol (RSVP)
- Routing Information Protocol (RIP)
- Simple Network Management Protocol (SNMP)

Personnel operating the equipment must be trained and competent; must not conduct themselves in a careless, willfully negligent, or hostile manner; and must abide by the instructions provided by the documentation.

Using the Indexes

This guide contains two indexes: a complete index that includes topic entries, and an index of statements and commands only.

In the index of statements and commands, an entry refers to a statement summary section only. In the complete index, the entry for a configuration statement or command contains at least two parts:

- The primary entry refers to the statement summary section.
- The secondary entry, *usage guidelines*, refers to the section in a configuration guidelines chapter that describes how to use the statement or command.

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the load merge or the load merge relative command. These commands cause the software to merge the incoming configuration into the current candidate configuration. If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example.* In this case, use the load merge command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the load merge relative command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file ex-script.conf. Copy the ex-script.conf file to the /var/tmp directory on your routing platform.

   ```
   system {
       scripts {
           commit {
               file ex-script.xsl;
           }
       }
   }
   interfaces {
       fxp0 {
           disable;
           unit 0 {
               family inet {
                   address 10.0.0.1/24;
               }
           }
       }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the load merge configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file ex-script-snippet.conf. Copy the ex-script-snippet.conf file to the /var/tmp directory on your routing platform.

   ```
   commit {
       file ex-script-snippet.xsl;
   }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

   ```
   [edit]
   user@host# edit system scripts
   [edit system scripts]
   ```

3. Merge the contents of the file into your routing platform configuration by issuing the load merge relative configuration mode command:

   ```
   [edit system scripts]
   user@host# load merge relative /var/tmp/ex-script-snippet.conf
   load complete
   ```

For more information about the load command, see the *JUNOS CLI User Guide*.

Documentation Conventions

Table 1 defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.

Table 2 defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Element	Example
Bold sans serif typeface	Represents text that you type.	To enter configuration mode, type the configure command: user@host> **configure**
`Fixed-width typeface`	Represents output on the terminal screen.	`user@host> show chassis alarms` `No alarms currently active`
Italic typeface	■ Introduces important new terms. ■ Identifies book names. ■ Identifies RFC and Internet draft titles.	■ A policy *term* is a named structure that defines match conditions and actions. ■ *JUNOS System Basics Configuration Guide* ■ RFC 1997, *BGP Communities Attribute*
Italic sans serif typeface	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Sans serif typeface	Represents names of configuration statements, commands, files, and directories; IP addresses; configuration hierarchy levels; or labels on routing platform components.	■ To configure a stub area, include the stub statement at the [edit protocols ospf area *area-id*] hierarchy level. ■ The console port is labeled CONSOLE.
< > (angle brackets)	Enclose optional keywords or variables.	stub <default-metric *metric*>;
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast | multicast (*string1* | *string2* | *string3*)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	community name members [*community-ids*]

Indention and braces ({ })	Identify a level in the configuration hierarchy.	[edit]
		routing-options {
		static {
		route default {
		nexthop *address*;
		retain;
		}
		}
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
J-Web GUI Conventions		
Bold typeface	Represents J-Web graphical user interface (GUI) items you click or select.	■ In the Logical Interfaces box, select **All Interfaces**. ■ To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select **Protocols > Ospf**.

Related Juniper Networks Documentation

Table 3 lists the software and hardware guides and release notes for Juniper Networks J-series, M-series, and T-series routing platforms and describes the contents of each document. Table 4 lists the books included in the *Network Operations Guide* series.

Table 3: Technical Documentation for J-series, M-series, and T-series Routing Platforms (1 of 3)

Document	Description
JUNOS Internet Software for J-series, M-series, and T-series Routing Platforms Configuration Guides	
Class of Service	Provides an overview of the class-of-service (CoS) functions of the JUNOS software and describes how to configure CoS features, including configuring multiple forwarding classes for transmitting packets, defining which packets are placed into each output queue, scheduling the transmission service level for each queue, and managing congestion through the random early detection (RED) algorithm.
CLI User Guide	Describes how to use the JUNOS command-line interface (CLI) to configure, monitor, and manage Juniper Networks routing platforms. This material was formerly covered in *JUNOS System Basics Configuration Guide.*
Feature Guide	Provides a detailed explanation and configuration examples for several of the most complex features in the JUNOS software.
JUNOS-FIPS	(M-series and T-series routing platforms only) Provides an overview of JUNOS-FIPS 140-2 concepts and describes how to install and configure the JUNOS-FIPS software. Describes FIPS-related commands and how to configure, authorize, and zeroize the Adaptive Services (AS) II FIPS Physical Interface Card (PIC).
MPLS Applications	Provides an overview of traffic engineering concepts and describes how to configure traffic engineering protocols.
Multicast Protocols	Provides an overview of multicast concepts and describes how to configure multicast routing protocols.
Network Interfaces	Provides an overview of the network interface functions of the JUNOS software and describes how to configure the network interfaces on the routing platform.
Network Management	Provides an overview of network management concepts and describes how to configure various network management features, such as SNMP and accounting options.
Policy Framework	Provides an overview of policy concepts and describes how to configure routing policy, firewall filters, forwarding options, and cflowd.
Routing Protocols	Provides an overview of routing concepts and describes how to configure routing, routing instances, and unicast routing protocols.
Services Interfaces	Provides an overview of the services interfaces functions of the JUNOS software and describes how to configure the services interfaces on the routing platform.
Software Installation and Upgrade Guide	Provides a description of JUNOS software components and packaging, and includes detailed information about how to initially configure, reinstall, and upgrade the JUNOS system software. This material was formerly covered in *JUNOS System Basics Configuration Guide.*
System Basics	Describes Juniper Networks routing platforms, and provides information about how to configure basic system parameters, supported protocols and software processes, authentication, and a variety of utilities for managing your router on the network.
VPNs	Provides an overview and describes how to configure Layer 2 and Layer 3 virtual private networks (VPNs), virtual private LAN service (VPLS), and Layer 2 circuits. Provides configuration examples.

Table 3: Technical Documentation for J-series, M-series, and T-series Routing Platforms (2 of 3)

Document	Description
JUNOS References	
Hierarchy and RFC Reference	Describes the JUNOS configuration mode commands. Provides a hierarchy reference that displays each level of a configuration hierarchy, and includes all possible configuration statements that can be used at that level. This material was formerly covered in *JUNOS System Basics Configuration Guide*.
Interfaces Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot interfaces.
Routing Protocols and Policies Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot routing protocols and policies, including firewall filters.
System Basics and Services Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot system basics, including commands for real-time monitoring and route (or path) tracing, system software management, and chassis management. Also describes commands for monitoring and troubleshooting services such as CoS, IP Security (IPSec), stateful firewalls, flow collection, and flow monitoring.
System Log Messages Reference	Describes how to access and interpret system log messages generated by JUNOS software modules and provides a reference page for each message.
J-Web User Guide	
J-Web Interface User Guide	Describes how to use the J-Web GUI to configure, monitor, and manage Juniper Networks routing platforms.
JUNOS API and Scripting Documentation	
JUNOScript API Guide	Describes how to use the JUNOScript application programming interface (API) to monitor and configure Juniper Networks routing platforms.
JUNOS XML API Configuration Reference	Provides reference pages for the configuration tag elements in the JUNOS XML API.
JUNOS XML API Operational Reference	Provides reference pages for the operational tag elements in the JUNOS XML API.
JUNOS Configuration and Diagnostic Automation Guide	Provides an overview, instructions for using, and examples of the commit script and self-diagnosis features of the JUNOS software. This guide explains how to enforce custom configuration rules defined in scripts that run at commit time, how to use commit script macros to provide simplified aliases for frequently used configuration statements, and how to configure diagnostic event policies and actions associated with each policy.
NETCONF API Guide	Describes how to use the NETCONF API to monitor and configure Juniper Networks routing platforms.
JUNOS Comprehensive Index and Glossary	
Comprehensive Index and Glossary	Provides a complete index of all JUNOS software books, the *JUNOScript API Guide*, and the *NETCONF API Guide*. Also provides a comprehensive glossary.
JUNOScope Documentation	
JUNOScope Software User Guide	Describes the JUNOScope software GUI, how to install and administer the software, and how to use the software to manage routing platform configuration files and monitor routing platform operations.
J-series Services Router Documentation	
J-series Services Router Getting Started Guide	Provides an overview, basic instructions, and specifications for J-series Services Routers. The guide explains how to prepare your site for installation, unpack and install the router and its components, install licenses, and establish basic connectivity.
J-series Services Router Basic LAN and WAN Access Configuration Guide	Explains how to configure the interfaces on J-series Services Routers for basic IP routing with standard routing protocols, ISDN backup, and digital subscriber line (DSL) connections.

Table 3: Technical Documentation for J-series, M-series, and T-series Routing Platforms (3 of 3)

Document	Description
J-series Services Router Advanced WAN Access Configuration Guide	Explains how to configure J-series Services Routers in virtual private networks (VPNs) and multicast networks, configure data link switching (DLSw) services, and apply routing techniques such as policies, stateless and stateful firewall filters, IP Security (IPSec) tunnels, and class-of-service (CoS) classification for safer, more efficient routing.
J-series Services Router Administration Guide	Shows how to manage users and operations, monitor network performance, upgrade software, and diagnose common problems on J-series Services Routers.
M-series and T-series Hardware Documentation	
Hardware Guide	Describes how to install, maintain, and troubleshoot routing platforms and components. Each platform has its own hardware guide.
PIC Guide	Describes the routing platform PICs. Each platform has its own PIC guide.
Release Notes	
JUNOS Release Notes	Summarize new features and known problems for a particular software release, provide corrections and updates to published JUNOS, JUNOScript, and NETCONF manuals, provide information that might have been omitted from the manuals, and describe upgrade and downgrade procedures.
Hardware Release Notes	Describe the available documentation for the routing platform and the supported PICs, and summarize known problems with the hardware and accompanying software. Each platform has its own release notes.
JUNOScope Software Release Notes	Contain corrections and updates to the published JUNOScope manual, provide information that might have been omitted from the manual, and describe upgrade and downgrade procedures.
J-series Services Router Release Notes	Briefly describe the J-series Services Router features, identify known hardware problems, and provide upgrade and downgrade instructions.

Table 4: JUNOS Internet Software Network Operations Guides

Book	Description
JUNOS Internet Software for M-series and T-series Routing Platforms Network Operations Guides	
Baseline	Describes the most basic tasks for running a network using Juniper Networks products. Tasks include upgrading and reinstalling JUNOS software, gathering basic system management information, verifying your network topology, and searching log messages.
Interfaces	Describes tasks for monitoring interfaces. Tasks include using loopback testing and locating alarms.
MPLS	Describes tasks for configuring, monitoring, and troubleshooting an example MPLS network. Tasks include verifying the correct configuration of the MPLS and RSVP protocols, displaying the status and statistics of MPLS running on all routers in the network, and using the layered MPLS troubleshooting model to investigate problems with an MPLS network.
MPLS Log Reference	Describes MPLS status and error messages that appear in the output of the show mpls lsp extensive command. The guide also describes how and when to configure Constrained Shortest Path First (CSPF) and RSVP trace options, and how to examine a CSPF or RSVP failure in a sample network.
Hardware	Describes tasks for monitoring M-series and T-series routing platforms.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. Send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at http://www.juniper.net/techpubs/docbug/docbugreport.html. If you are using e-mail, be sure to include the following information with your comments:

- Document name
- Document part number
- Page number
- Software release version

Requesting Support

For technical support, open a support case using the Case Manager link at http://www.juniper.net/support/ or call 1-888-314-JTAC (from the United States, Canada, or Mexico) or 1-408-745-9500 (from elsewhere).

Part 1
Introduction

Chapter 1

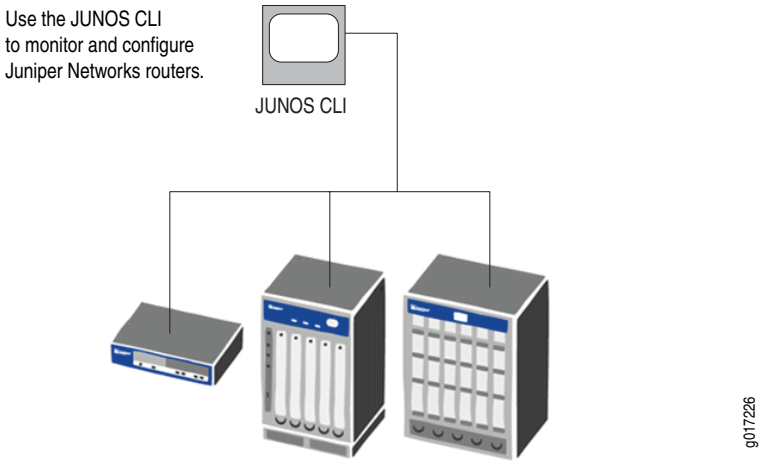
Introducing the CLI

This chapter provides an overview of the JUNOS software command-line interface (CLI). The CLI is the software interface you use to access the router—whether from the console or through a remote network connection.

The JUNOS CLI is a Juniper-specific command shell that runs on top of a UNIX-based operating system kernel. By leveraging industry standard tools and utilities, the CLI provides you with a powerful set of commands you can use to monitor and configure a router. (See Figure 1.)

The CLI is a straightforward command interface. You type commands on a single line, and the commands are executed when you press the Enter key. The CLI provides command help and command completion, and it also provides Emacs-style keyboard sequences that allow you to move around on a command line and scroll through recently executed commands.

Figure 1: Monitoring and Configuring Routers



Topics in this chapter include:

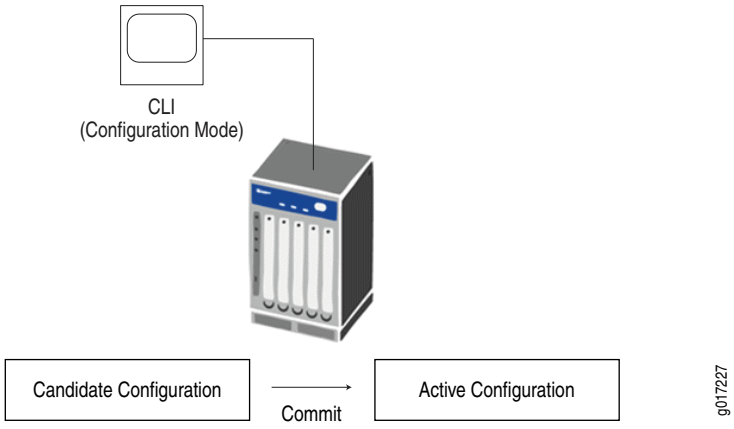
Understanding CLI Command Modes

The CLI has two modes:

- Operational mode—Displays the current router status. In operational mode, you enter commands to monitor and troubleshoot the software, network connectivity, and router.
- Configuration mode—A router configuration is stored as a hierarchy of statements. In configuration mode, you enter these statements to define all properties of the JUNOS software, including interfaces, general routing information, routing protocols, user access, and several system hardware properties.

When you enter configuration mode, you are actually viewing and changing a file called the *candidate configuration*. The candidate configuration allows you to make configuration changes without causing operational changes to the current operating configuration, called the *active configuration*. The router does not implement the changes you added to the candidate configuration until you commit them, which activates the configuration on the router. (See Figure 2.) Candidate configurations enable you to alter your configuration without causing potential damage to your current network operations.

Figure 2: Committing a Configuration



Understanding Command and Statement Hierarchies

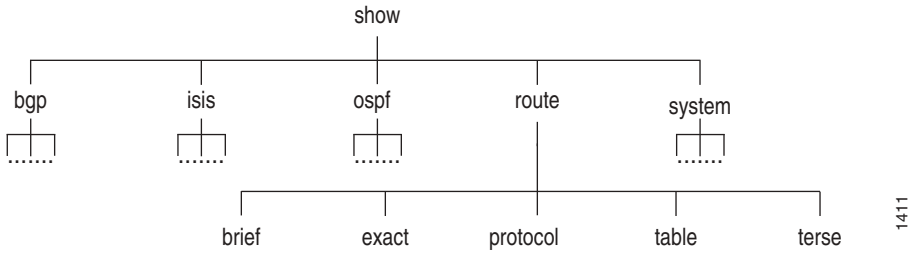
The JUNOS software provides a hierarchal organization for CLI commands and a hierarchal organization for configuration statements.

Topics in this section include:

CLI Command Hierarchy

CLI commands are organized in a hierarchy. Commands that perform a similar function are grouped together under the same level of the hierarchy. For example, all commands that display information about the system and the system software are grouped under the show system command, and all commands that display information about the routing table are grouped under the show route command. Figure 3 illustrates a portion of the show command hierarchy.

Figure 3: CLI Command Hierarchy Example



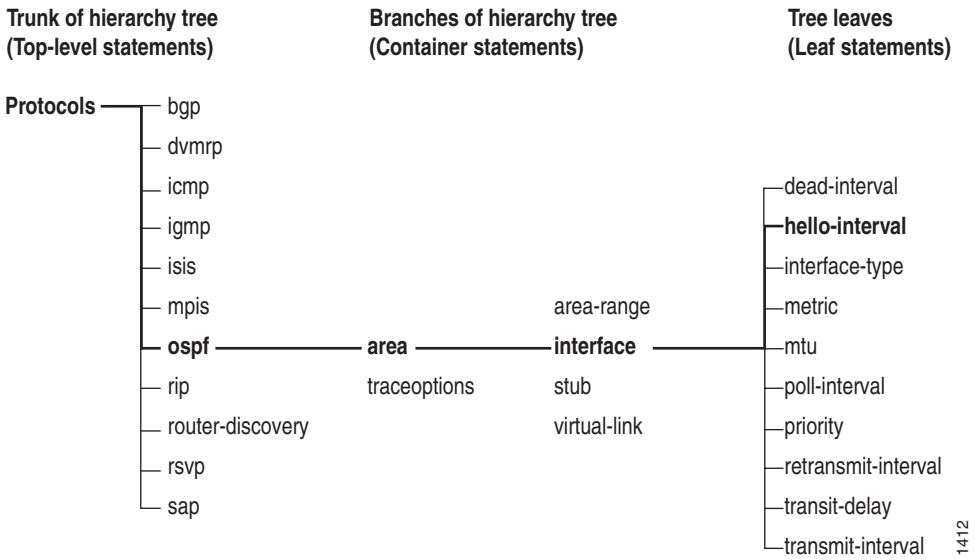
To execute a command, you enter the full command name, starting at the top level of the hierarchy. For example, to display a brief view of the routes in the router table, use the command show route brief.

Configuration Statement Hierarchy

The configuration statement hierarchy has two types of statements: *container statements*, which are statements that contain other statements, and *leaf statements*, which do not contain other statements. All of the container and leaf statements together form the *configuration hierarchy*.

Figure 4 on page 6 illustrates a part of the hierarchy tree. The protocols statement is a top-level statement at the trunk of the configuration tree. The ospf, area, and interface statements are all subordinate container statements of a higher statement (they are branches of the hierarchy tree), and the hello-interval statement is a leaf on the tree.

Figure 4: Configuration Statement Hierarchy Example



Key Features of the CLI

The hierarchical organization results in commands that have a regular syntax and provides several features that simplify CLI use:

- Consistent command names—Commands that provide the same type of function have the same name, regardless of the portion of the software on which they are operating. As examples, all show commands display software information and statistics, and all clear commands erase various types of system information.

- Lists and short descriptions of available commands—Information about available commands is provided at each level of the CLI command hierarchy. If you type a question mark (?) at any level, you see a list of the available commands along with a short description of each command. This means that if you already are familiar with the JUNOS software or with other routing software, you can use many of the CLI commands without referring to the documentation.

- Command completion—Command completion for command names (keywords) and for command options is also available at each level of the hierarchy. To complete a command or option that you have partially typed, press the tab key or the spacebar. If the partially typed letters begin a string that uniquely identifies a command, the complete command name appears. Otherwise, a beep indicates that you have entered an ambiguous command, and the possible completions are displayed. Command completion also applies to other strings, such as filenames, interface names, and usernames.

Leveraging Industry-Standard Technologies

The Juniper Networks operating system is based on a FreeBSD UNIX kernel, with a special shell called the CLI (command-line interface). With FreeBSD UNIX as the kernel, a variety of UNIX utilities are available on the router. For example, you can:

- Use regular expression matching to locate and replace values and identifiers in a configuration, filter command output, or examine log file entries.
- Use Emacs-based key sequences to scroll through command output or edit the command line,
- Store and archive router files on a UNIX-based file system.
    - You can use standard UNIX conventions to specify filenames and paths.
    - You can exit from the CLI environment and create a UNIX C shell or Bourne shell to navigate the file system, manage router processes, and so on.

Other Tools to Configure and Monitor JUNOS Routers

The JUNOS software also supports the following applications, scripts, and utilities you can use to configure and monitor JUNOS routers:

- J-Web graphical user interface (GUI)—Allows you to monitor, configure, troubleshoot, and manage the router on a client by means of a Web browser with Hypertext Transfer Protocol (HTTP) or HTTP over Secure Sockets Layer (HTTPS) enabled. For more information, see the *J-Web Interface User Guide*.
- JUNOScript Application Programming Interface (API)—Application programmers can use the JUNOScript API to monitor and configure Juniper Networks routing platforms. Juniper Networks provides a Perl module with the API to help you more quickly and easily develop custom Perl scripts for configuring and monitoring routing platforms. For more information, see the *JUNOScript API Guide*.
- NETCONF Application Programming Interface (API)—Application programmers can also use the NETCONF API to monitor and configure Juniper Networks routing platforms. For more information, see the *NETCONF API Guide*.
- JUNOS commit scripts and self-diagnosis features—You can define scripts to enforce custom configuration rules, use commit script macros to provide simplified aliases for frequently used configuration statements, and configure diagnostic event policies and actions associated with each policy. For more information, see the *JUNOS Configuration and Diagnostic Automation Guide*.
- Management Information Bases (MIBs)—You can use enterprise-specific and standard MIBS to retrieve management information for various hardware and software components on a Juniper Networks router. For more information about MIBS, see the *JUNOS Network Management Configuration Guide*.

Commands and Configuration Statements for JUNOS-FIPS

JUNOS-FIPS enables you to configuring a network of Juniper Networks routers in a Federal Information Processing Standards (FIPS) 140-2 environment.

The JUNOS-FIPS software environment requires the installation of FIPS software by a Crypto Officer. In JUNOS-FIPS, some JUNOS commands and statements have restrictions and some additional configuration statements are available. For more information, see the *JUNOS-FIPS Configuration Guide.*

Chapter 2

Getting Started: A Quick Tour of the CLI

As an introduction to the command-line interface (CLI), this chapter provides instructions for simple steps you take after installing software on the router. It shows you how to start the CLI, view the command hierarchy, and make small configuration changes. Detailed information about using the CLI appears in subsequent chapters.

Topics include:

NOTE: The instructions and examples in this book are based on M-series and T-series routers, but you can use them as a guideline for entering commands on your own router.

Before You Start

Make sure your router hardware is set up and the latest JUNOS software is installed. You must have a direct console connection to the router or network access via SSH or Telnet. If your router is not set up, follow the installation instructions provided with your router before proceeding.

Logging In

Log in to the router and start the CLI:

1. Log in as root.

   The root login account has superuser privileges, with access to all commands and statements.

2. Start the CLI:

   ```
   root# cli
   root@>
   ```

   The > command prompt shows you are in operational mode. Later, when you enter configuration mode, the prompt will change to #.

NOTE: If you are using the root account for the first time on the router, remember that the router ships with no password required for root, but the first time you commit a configuration with JUNOS 7.6 software, you must set a root password. Root access is not allowed over a telnet session. To enable root access over an SSH connection, you must configure the system services ssh root-login allow statement.

Displaying Commands

1. Type **?** to show the top-level commands available in operational mode.

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

   The CLI includes several ways to get help on commands. Typing a ? is one example. For more information, see “Getting Online Help” on page 33.

Checking Router Status

You can use show commands to check router status and monitor router activity.

To help you become familiar with show commands:

1. Use the show chassis routing-engine command to view Routing Engine status:

```
root@> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 31 degrees C / 87 degrees F
    CPU temperature             32 degrees C / 89 degrees F
    DRAM                       768 MB
    Memory utilization          84 percent
    CPU utilization:
      User                      0 percent
      Background                0 percent
      Kernel                    1 percent
      Interrupt                 0 percent
      Idle                     99 percent
    Model                       RE-2.0
    Serial ID                   b10000078c10d701
    Start time                  2005-12-28 13:52:00 PST
    Uptime                     12 days, 3 hours, 44 minutes, 19 seconds
    Load averages:              1 minute   5 minute  15 minute
                                    0.02       0.01       0.00
```

2. Use the show system storage command to view available storage on the router:

```
root@> show system storage
Filesystem              Size       Used      Avail  Capacity   Mounted on
/dev/ad0s1a              78M        47M        25M       65%  /
devfs                    16K        16K         0B      100%  /dev/
/dev/vn0                 14M        14M         0B      100%
/packages/mnt/jbase
/dev/vn1                 52M        52M         0B      100%
/packages/mnt/jkernel-7.6R1.0
/dev/vn2               10.0M      10.0M         0B      100%
/packages/mnt/jpfe-M40-7.6R1.0
/dev/vn3                2.5M       2.5M         0B      100%
/packages/mnt/jdocs-7.6R1.0
/dev/vn4                 17M        17M         0B      100%
/packages/mnt/jroute-7.6R1.07.6R1.0
/dev/vn5                6.7M       6.7M         0B      100%
/packages/mnt/jcrypto-7.6R1.0
/dev/vn6                7.1M       7.1M         0B      100%
/packages/mnt/jpfe-common-7.6R1.0
mfs:136                 1.5G        60K       1.4G        0%  /tmp
mfs:148                 1.5G       1.2M       1.3G        0%  /mfs
/dev/ad0s1e              12M        26K        11M        0%  /config
procfs                  4.0K       4.0K         0B      100%  /proc
/dev/ad1s1f             9.4G       1.1G       7.5G       13%  /var
/dev/vn7                5.5M       5.5M         0B      100%
/packages/mnt/jtools-7.6R1.0
```

Configuring a User Account

This section describes how to log in with a root account and configure a new user account. You can configure an account for your own use or create a test account.

To configure a user account on the router:

1. Log in as root and enter configuration mode:

   ```
   root@host> configure
   [edit]
   root@host#
   ```

   The prompt in brackets ([edit]) shows that you are in configuration edit mode, at the top of the hierarchy.

2. Change to the [edit system login] section of the configuration:

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

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

   The prompt in brackets changes to [edit system login] to show you are at a new level in the hierarchy.

3. Now add a new user account:

   ```
   [edit system login]
   root@host# edit user nchen
   ```

   This example shows how to add an account for nchen (Nathan Chen), but you can use your own login name if desired.

4. Configure a full name for the account. If the name includes spaces, enclose the entire name in quotation marks (" "):

   ```
   [edit system login user nchen]
   root@host# set full-name "Nathan Chen"
   ```

5. Configure an account class. The account class sets the user access privileges for the account.

   ```
   [edit system login user nchen]
   root@host# set class super-user
   ```

6. Configure an authentication method and password for the account:

   ```
   [edit system login user nchen
   root@host# set authentication plain-text-password
   New password:
   Retype new password:
   ```

   When the new password prompt appears, enter a clear-text password that the system will encrypt, then confirm the new password.

7. Commit the configuration:

   ```
   [edit system login user nchen]
   root@host# commit
   commit complete
   ```

   Configuration changes are not activated until you commit the configuration. If the commit is successful, a commit complete message appears.

8. Return to the top level of the configuration, then exit:

   ```
   [edit system login user nchen]
   root@host# top

   [edit]
   root@host# exit
   Exiting configuration mode
   ```

9. Log out of the router:

   ```
   root@host> exit

   % logoutConnection closed.
   ```

10. To test your changes, log back in with the user account and password you just configured:

    ```
    >
    login: nchen
    Password: <password>

    — JUNOS 7.6-R1.1 built 2005-12-15 22:42:19 UTC
    nchen@host>
    ```

    When you log in, you should see the new username at the command prompt.

Congratulations! You have successfully used the CLI to view router status and perform a simple configuration change. Now, you are ready to learn more about the CLI. See the remaining chapters of this book for details.

Alternatively, you can follow the instructions in “Configuring Interfaces” on page 14 to create a more extensive configuration.

NOTE: For complete information about the commands to issue to configure your router, including examples, see the JUNOS software configuration guides.

Configuring a Routing Protocol

This section describes how to conifigure an OSPF routing protocol. You can use it as a starting point for configuring additional protocols at a later time.

The example used in this section creates the following configuration:

```
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0 {
                hello-interval 5;
                dead-interval 20;
            }
            interface so-0/0/1 {
                hello-interval 5;
                dead-interval 20;
            }
        }
    }
}
```

Shortcut

You can create this entire configuration with two commands:

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/0 hello-interval 5
dead-interval 20
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/1 hello-interval 5
dead-interval 20
```

Longer Configuration Example

This section provides a longer example of creating the above OSPF configuration. In the process, it illustrates how to use the different features of the CLI.

1. Enter configuration mode by issuing the configure top-level command:

   ```
   user@host> configure
   entering configuration mode
   [edit]
   user@host#
   ```

   Notice that the prompt has changed to a pound sign (#) to indicate configuration mode.

2. To create the above configuration, you start by editing the protocols ospf statements:

   ```
   [edit]
   user@host# edit protocols ospf
   [edit protocols ospf]
   user@host#
   ```

3. Now, add the OSPF area:

```
[edit protocols ospf]
user@host# edit area 0.0.0.0
[edit protocols ospf area 0.0.0.0]
user@host#
```

4. Add the first interface:

```
[edit protocols ospf area 0.0.0.0]
user@host# edit interface so0
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host#
```

You now have four nested statements.

5. Set the hello and dead intervals.

Remember that command completion (enter a tab or space) and context-sensitive help (type a question mark) are always available.

```
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host# set ?
Possible completions:
+ apply-groups         Groups from which to inherit configuration data
> authentication-key   Authentication key
  dead-interval        Dead interval (seconds)
  disable              Disable OSPF on this interface
  hello-interval       Hello interval (seconds)
  interface-type       Type of interface
  metric               Interface metric (1..65535)
> neighbor             NBMA neighbor
  passive              Do not run OSPF, but advertise it
  poll-interval        Poll interval for NBMA interfaces
  priority             Designated router priority
  retransmit-interval  Retransmission interval (seconds)
  transit-delay        Transit delay (seconds)
  transmit-interval    OSPF packet transmit interval (milliseconds)
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host# set hello-interval 5
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host# set dead-interval 20
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host#
```

6. You can see what is configured at the current level with the show command:

```
[edit protocols ospf area 0.0.0.0 interface so-o]
user@host# show
hello-interval 5;
dead-interval 20;
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host#
```

7. You are finished at this level, so back up a level and take a look at what you have so far:

```
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host# up
[edit protocols ospf area 0.0.0.0]
user@host# show
interface so-0/0/0 {
    hello-interval 5;
    dead-interval 20;
}
[edit protocols ospf area 0.0.0.0]
user@host#
```

   The interface statement appears because you have moved to the area statement.

8. Add the second interface:

```
[edit protocols ospf area 0.0.0.0]
user@host# edit interface so-0/0/1
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# set hello-interval 5
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# set dead-interval 20
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# up
[edit protocols ospf area 0.0.0.0]
user@host# show
interface so-0/0/0 {
    hello-interval 5;
    dead-interval 20;
}
interface so-0/0/1 {
    hello-interval 5;
    dead-interval 20;
}
[edit protocols ospf area 0.0.0.0]
user@host#
```

9. Back up to the top level and see what you have:

```
[edit protocols ospf area 0.0.0.0]
user@host# top
[edit]
user@host# show
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0 {
                hello-interval 5;
                dead-interval 20;
            }
            interface so-0/0/1 {
                hello-interval 5;
                dead-interval 20;
            }
        }
    }
}
[edit]
user@host#
```

   This configuration now contains the statements you want.

10. Before committing the configuration (and activating it), verify that the configuration is correct:

```
[edit]
user@host# commit check
configuration check succeeds
[edit]
user@host#
```

11. Commit the configuration to activate it on the router:

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

   Suppose you decide to use different dead and hello intervals on interface so-0/0/1. You can make changes to the configuration.

12. You can go directly to the appropriate hierarchy level by typing the full hierarchy path to the statement you want to edit.

```
[edit]
user@host# edit protocols ospf area 0.0.0.0 interface so-0/0/1
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# show
hello-interval 5;
dead-interval 20;
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# set hello-interval 7
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# set dead-interval 28
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# top
[edit]
user@host# show
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0 {
                hello-interval 5;
                dead-interval 20;
            }
            interface so-0/0/1 {
                hello-interval 7;
                dead-interval 28;
            }
        }
    }
}
[edit]
user@host#
```

13. If you change your mind and decide not to run OSPF on the first interface, you can delete the statement:

```
[edit]
user@host# edit protocols ospf area 0.0.0.0
[edit protocols ospf area 0.0.0.0]
user@host# delete interface so-0/0/0
[edit protocols ospf area 0.0.0.0]
user@host# top
[edit]
user@host# show
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/1 {
                hello-interval 7;
                dead-interval 28;
            }
        }
    }
}
[edit]
user@host#
```

Everything inside the statement you deleted was deleted with it. You can also eliminate the entire OSPF configuration by simply entering delete protocols ospf while at the top level.

14. Suppose you decide to use the default values for the hello and dead intervals on your remaining interface, but you want OSPF to run on that interface:

```
[edit]
user@host# edit protocols ospf area 0.0.0.0 interface so-0/0/1
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# delete hello-interval
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# delete dead-interval
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# top
[edit]
user@host# show
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/1;
        }
    }
}
[edit]
user@host#
```

You can set multiple statements at the same time as long as they are all part of the same hierarchy (the path of statements from the top inward, as well as one or more statements at the bottom of the hierarchy). This can reduce considerably the number of commands you must enter.

15. To go back to the original hello and dead interval timers on interface so-0/0/1, you can enter:

```
[edit]
user@host# edit protocols ospf area 0.0.0.0 interface so-0/0/1
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# set hello-interval 5 dead-interval 20
[edit protocols ospf area 0.0.0.0 interface so-0/0/1]
user@host# exit
[edit]
user@host# show
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/1 {
                hello-interval 5;
                dead-interval 20;
            }
        }
    }
}
[edit]
user@host#
```

16. You also can recreate the other interface, as you had it before, with only a single entry:

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/1 hello-interval 5
dead-interval 20
[edit]
user@host# show
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0 {
                hello-interval 5;
                dead-interval 20;
            }
            interface so-0/0/1 {
                hello-interval 5;
                dead-interval 20;
            }
        }
    }
}
```

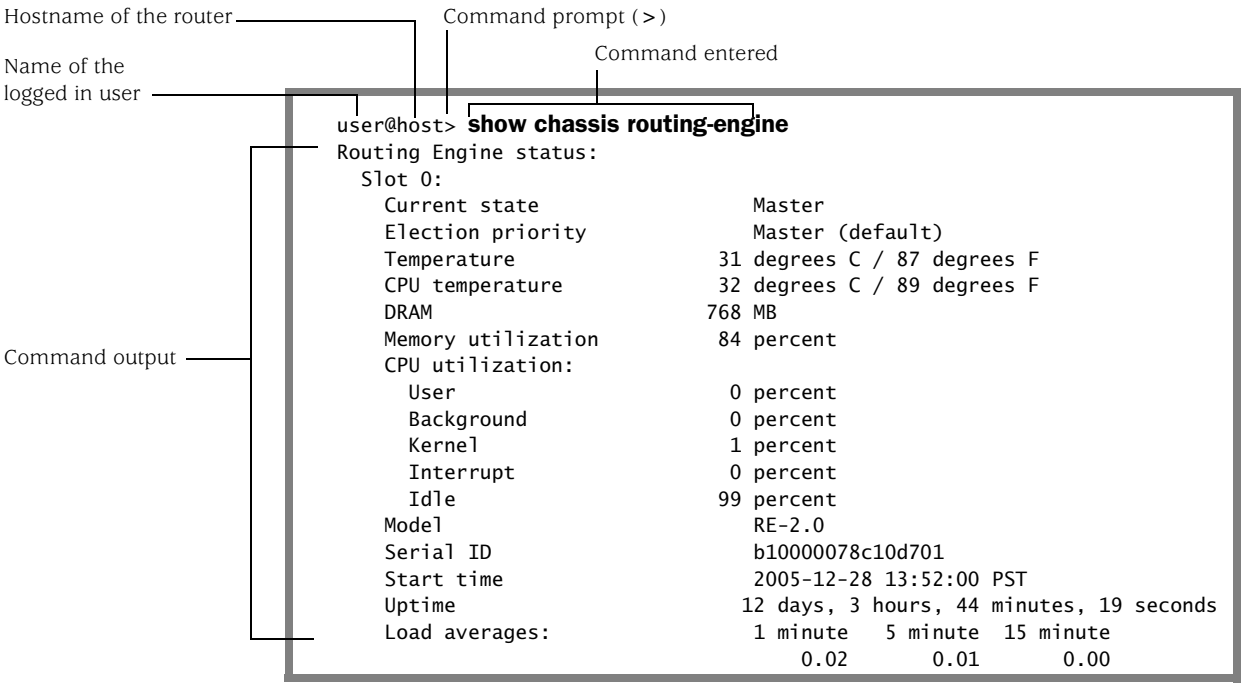
Chapter 3
CLI Basics

This chapter provides basic information about the CLI. Topics include:

Elements of the Command-Line Interface

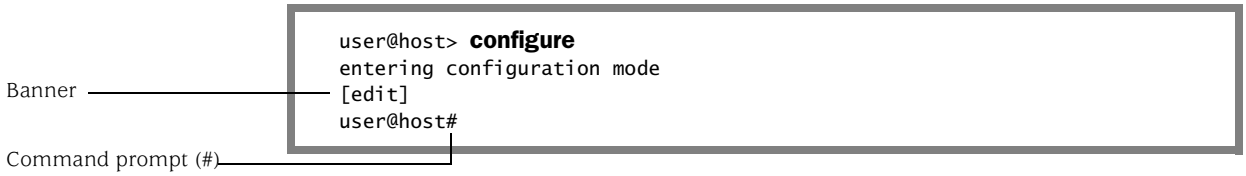
Elements of the command-line interface in operational mode are shown in Figure 5.

Figure 5: Elements of the Command-Line Interface



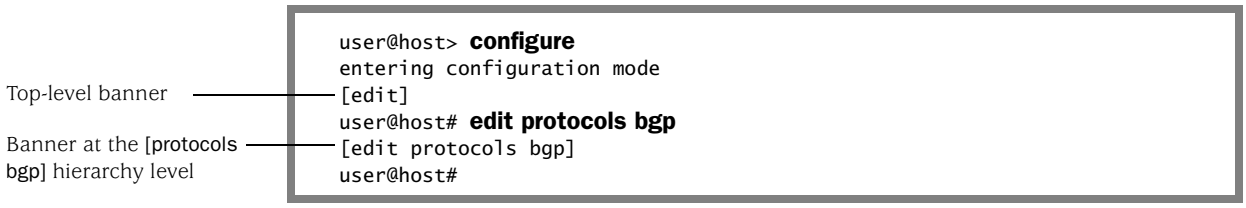
In configuration mode, the prompt changes from a > to a #. (See Figure 6.)

Figure 6: Command Prompt in Configuration Mode



The portion of the prompt in square brackets, [edit], is a *banner*. The banner indicates that you are in configuration mode and shows your location in the statement hierarchy. When you first enter configuration mode, you are always at the top level of the hierarchy, as indicated by the [edit] banner. (See Figure 7.)

Figure 7: Hierarchy Level Banner



When you change to configuration mode, the command prompt also changes. The operational mode prompt is a right angle bracket (>) and the configuration mode prompt is a pound sign (#).

CLI Messages

Messages appear when you enter and exit from command mode, when you successfully complete some commands, and when you type an invalid string or value.

If you type an invalid string—for example, the name of a command or statement that does not exist—you see the message "syntax error" or "unknown command". A caret (^) indicates where the error is. For example:

```
user@host> clear route
                     ^
syntax error, expecting <command>.

[edit]
user@host# telnet
           ^
unknown command.
```

When the number of choices is limited, a message might display the commands you can enter to correct the syntax error. For example:

```
[edit]
user@host# load myconfig-file<Enter>
               ^
syntax error, expecting 'merge', 'override', or 'replace'.
```

In configuration mode, if you do not type an option for a statement that requires one, a message indicates the type of information expected.

In this example, you need to type an area number to complete the command:

```
[edit]
user@host# set protocols ospf area<Enter>
                                  ^
syntax error, expecting <identifier>.
```

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

```
[edit]
user@host# set protocols ospf area 45 interface so-0/0/0
           hello-interval<Enter>
                                 ^
syntax error, expecting <data>
```

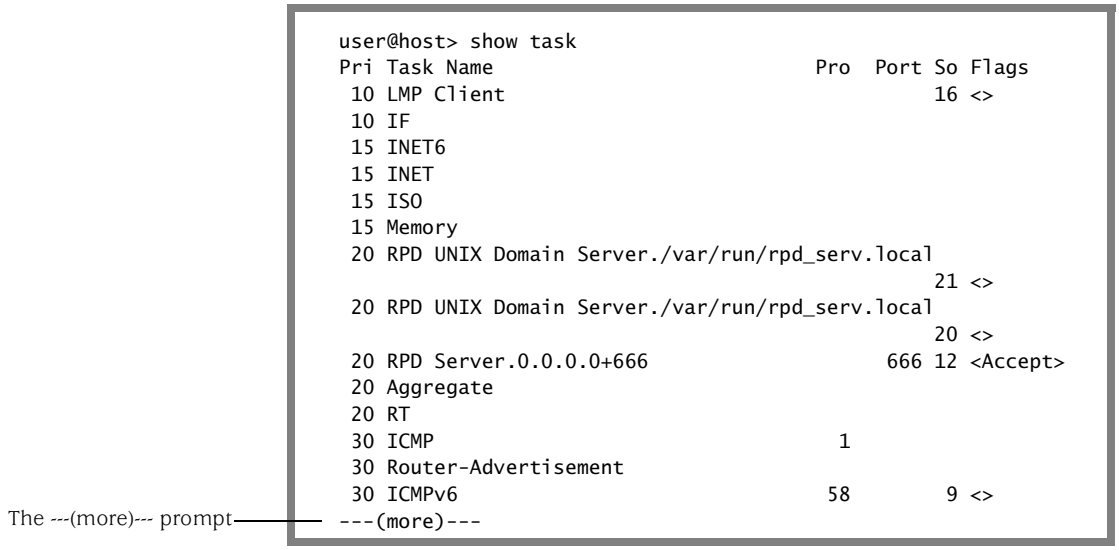
If you have omitted a required statement at a particular hierarchy level, when you attempt to move from that hierarchy level or when you issue the show command in configuration mode, a message indicates which statement is missing. For example:

```
[edit system login user phil]
user@host# up
Warning: missing mandatory statement: 'class'
[edit system login]
user@host# show
user phil {
        full-name “Phil James”;
           # Warning: missing mandatory statement(s): 'class'
}
```

Displaying Command Output

If the command output is longer than the screen length, it appears one screen at a time by means of a UNIX more-type interface. The prompt —(more)— indicates that more output is available. (See Figure 8.)

Figure 8: The —(more)— Prompt



To continue command output, press Return.

Table 5 lists some common keyboard sequences you can use at the —(more)— prompt. As soon as the CLI knows how long the output is (usually by the second screen), it displays the percentage of the command output above the prompt.

Occasionally, if a command produces extensive output, you may wish to cancel the output.

To cancel command output, press Ctrl+c. Command output stops and the command prompt appears. (See Figure 9.)

Figure 9: Cancelling Command Output

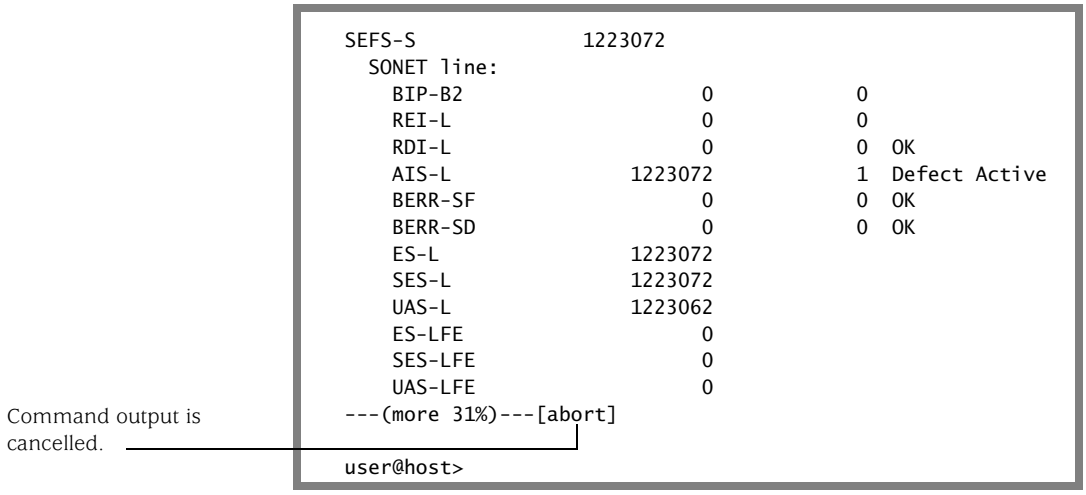


Table 5: —more— Prompt Keyboard Sequences

Category	Action	Keyboard Sequence
Scroll Down	Scroll down one line.	Enter, Return, k, Ctrl+m, Ctrl+n, or down arrow
	Scroll down one-half screen.	Tab, d, Ctrl+d, or Ctrl+x
	Scroll down one whole screen.	Space or Ctrl+f
	Scroll down to the bottom of the output.	Ctrl+e or G
	Display the output all at once instead of one screen at a time. (Same as specifying the | no-more command.)	N
Scroll Up	Display the previous line of output.	j, Ctrl+h, Ctrl+p, or up arrow
	Scroll up one-half screen.	u or Ctrl+u
	Scroll up one whole screen.	b or Ctrl+b
	Scroll up to the top of the output.	Ctrl+a or g

For more information about working with command output, see “Filtering Command Output” on page 127.

Types of Commands and Statements

The JUNOS CLI supports the following types of commands and statements:

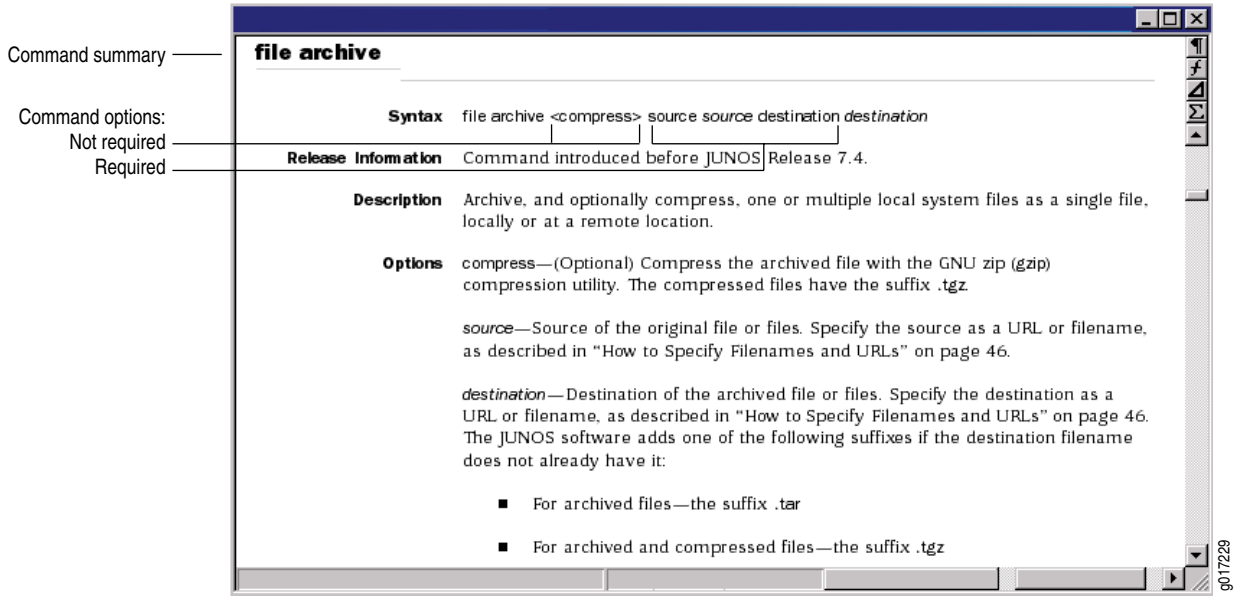
- CLI operational mode commands—Commands you enter in operational mode are used to monitor router operation. This book describes the top-level operational mode commands.
  - For more information on using top-level CLI operational mode commands, see "Using CLI Operational Commands to Monitor the Router" on page 45.
  - For a complete list of CLI operational mode commands, see the JUNOS command reference guides.
- CLI environment commands—A set of operational mode commands you can use to control the CLI environment. For example, you can specify the number lines that are displayed on the screen or your terminal type. For more information, see "Controlling the CLI Environment" on page 137.
- CLI configuration mode commands—Commands you enter in configuration mode are used to perform general configuration functions; for example, committing a configuration, copying statements, navigating the hierarchy, and managing configuration files. For more information, see "Using Commands and Statements to Configure the Router" on page 71.
- Configuration statements—Used to define your router configuration. Your location in the configuration hierarchy determines which configuration statements are available. For example, the [edit interfaces] hierarchy level includes statements to configure router interfaces.
  - For more information, see the JUNOS software configuration guides.
  - For a complete list of JUNOS configuration statements and the statement hierarchy, see the *JUNOS Hierarchy and RFC Reference*.

Command Options

When working on the command line, you are bound by certain syntax rules. Some commands function very simply with just a single word necessary to run them. Others have required options that you must enter to complete the command. Additionally, commands may have options which are not required, allowing you to change the way they run or the information they return.

The command summaries in the JUNOS software books show which options are required and which options are not required. Options that are not required are shown with angle brackets (< >). (See Figure 10 on page 28.)

Figure 10: Command Options



Configuration Statements and Identifiers

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

The following list shows the statements available at the top level of configuration mode (that is, the trunk of the hierarchy tree).

```
user@host# set ?
Possible completions:
> accounting-options     Accounting data configuration
+ apply-groups           Groups from which to inherit configuration data
> chassis                Chassis configuration
> class-of-service       Class-of-service configuration
> firewall               Define a firewall configuration
> forwarding-options     Configure options to control packet sampling
> groups                 Configuration groups
> interfaces             Interface configuration
> policy-options         Routing policy option configuration
> protocols              Routing protocol configuration
> routing-instances      Routing instance configuration
> routing-options        Protocol-independent routing option configuration
> snmp                   Simple Network Management Protocol
> system                 System parameters
```

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

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

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

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

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

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

Privilege Levels

Each top-level CLI command and each configuration statement has an access privilege level associated with it. Users can execute only those commands and configure and view only those statements for which they have access privileges.

For example, users with configure permissions can use the configure command to enter configuration mode, and users with firewall permissions can view the firewall filter configuration in configuration mode. The root login account has superuser privileges—with access to all commands and statements.

Required privilege levels are listed in command and statement summaries. For more information on setting user accounts and privileges, see the *JUNOS System Basics Configuration Guide*.

Switching Between Operational Mode and Configuration Mode

When you monitor and configure a router, you may need to switch between operational mode and configuration mode.

To switch between operational mode and configuration mode:

1. When you log in to the router and type the cli command, you are automatically in operational mode:

   ```
   user@host# cli
   user@host >
   ```

2. To enter configuration mode, type the configure command or the edit command from the CLI operation mode. For example:

   ```
   user@host> configure
   entering configuration mode
   [edit]
   ```

   The CLI prompt changes from user@host> to user@host# and a banner appears to indicate the hierarchy level.

3. To return to operational mode, you must exit (and optionally commit) the configuration.

   To commit the configuration and exit:

   ```
   [edit]
   user@host# commit and-quit
   commit complete
   Exiting configuration mode
   user@host>
   ```

   To exit without committing:

   ```
   [edit]
   user@host# exit
   Exiting configuration mode
   user@host>
   ```

   When you exit configuration mode, the CLI prompt changes from user@host# to user@host> and the banner no longer appears. You may enter or exit configuration mode as many times as you wish without committing your changes.

   At times, you might need to display the output of an operational mode show or other command while configuring the software.

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

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

For example, to display the priority value of the Virtual Router Redundancy Protocol (VRRP) primary router while you are modifying the VRRP configuration for a backup router:

```
[edit interfaces ge-4/2/0 unit 0 family inet vrrp-group 27]
user@host# show
virtual-address [ 192.168.1.15 ];
[edit interfaces ge-4/2/0 unit 0 family inet vrrp-group 27]
user@host# run show vrrp detail
Physical interface: ge-5/2/0, Unit: 0, Address: 192.168.29.10/24
  Interface state: up, Group: 10, State: backup
  Priority: 190, Advertisement interval: 3, Authentication type: simple
  Preempt: yes, VIP count: 1, VIP: 192.168.29.55
  Dead timer: 8.326, Master priority: 201, Master router: 192.168.29.254
[edit interfaces ge-4/2/0 unit 0 family inet vrrp-group 27]
user@host# set priority ...
```

Moving Among Hierarchy Levels

The CLI commands in Table 6 help you navigate the levels of the configuration statement hierarchy.

Table 6: CLI Configuration Mode Navigation Commands

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

Displaying CLI Command History

You can display a list of recent commands that you issued. To display the command history, use the show cli history command:

```
user@host> show cli history
  03-03 01:00:50 – show cli history
  03-03 01:01:12 – show interfaces terse
  03-03 01:01:22 – show interfaces lo0
  03-03 01:01:44 – show bgp next-hop-database
  03-03 01:01:51 – show cli history
```

By default, this command displays the last 100 commands issued in the CLI. If you specify a number with the command, it displays that number of recent commands. For example:

```
user@host> show cli history 3
  01:01:44 – show bgp next-hop-database
  01:01:51 – show cli history
  01:02:51 – show cli history 3
```

Displaying CLI Word History

You can type Esc+. or Alt+. to insert the last word of the previous command. Repeat Esc+. or Alt+. to scroll backwards through the list of recently entered words. For example:

```
user@host> show interfaces terse fe-0/0/0
Interface               Admin   Link   Proto  Local                 Remote
fe-0/0/0                up      up
fe-0/0/0.0              up      up     inet   192.168.220.1/30
user@host> fe-0/0/0
```

If you scroll completely to the beginning of the list, typing Esc+. or Alt+. again restarts scrolling from the last word entered.

Chapter 4

Getting Online Help

This chapter provides basic information about getting online help for CLI commands and statements.

Topics include:

Getting Help About Commands

Information about commands is provided at each level of the CLI command hierarchy. You can type a question mark to get help about commands:

- If you type the question mark at the command-line prompt, the CLI lists the available commands and options. For example, to view a list of top-level operational mode commands, type a question mark (?) at the command-line prompt.

```
user@host> ?
Possible completions:
  clear          Clear information in the system
  configure      Manipulate software configuration information
  file           Perform file operations
  help           Provide help information
  mtrace         Trace mtrace packets from source to receiver.
  monitor        Real-time debugging
  ping           Ping a remote target
  quit           Exit the management session
  request        Make system-level requests
  restart        Restart a software process
  set            Set CLI properties, date, time, craft display text
  show           Show information about the system
  ssh            Open a secure shell to another host
  start          Start a software process
  telnet         Telnet to another host
  test           Diagnostic debugging commands
  traceroute     Trace the route to a remote host
user@host>
```

- If you type the question mark after entering the complete name of a command or command option, the CLI lists the available commands and options, then redisplays the command names and options that you typed.

```
user@host> clear ?
Possible completions:
  arp            Clear address-resolution information
  bgp            Clear BGP information
  chassis        Clear chassis information
  firewall       Clear firewall counters
  igmp           Clear IGMP information
  interfaces     Clear interface information
  ilmi           Clear ILMI statistics information
  isis           Clear IS-IS information
  ldp            Clear LDP information
  log            Clear contents of a log file
  mpls           Clear MPLS information
  msdp           Clear MSDP information
  multicast      Clear Multicast information
  ospf           Clear OSPF information
  pim            Clear PIM information
  rip            Clear RIP information
  route          Clear routing table information
  rsvp           Clear RSVP information
```

```
  snmp          Clear SNMP information
  system        Clear system status
  vrrp          Clear VRRP statistics information
user@host> clear
```

- If you type the question mark in the middle of a command name, the CLI lists possible command completions that match the letters you have entered so far, then redisplays the letters that you typed. For example, to list all operational mode commands that start with the letter c, type the following:

  ```
  user@host> c?
  Possible completions:
    clear           Clear information in the system
    configure       Manipulate software configuration information
  user@host>c
  ```

- For introductory information on using the question mark or the help command, you can also type **help** and press Enter:

  help

Getting Help Based on a String in a Statement Name

You can use the help command to display help based on a text string contained in a statement name. This command displays help for statements at the current hierarchy level and below:

help apropos *string*

string is a text string about which you want to get help. This string is used to match statement names as well as the help strings that are displayed for the statements. (See the example on page 36.)

If the string contains spaces, enclose it in quotation marks (" "). You also can specify a regular expression for the string, using standard UNIX-style regular expression syntax.

You can display help based on a text string contained in a statement name using the help topic and help reference commands:

help topic *string*
help reference *string*

The help topic command displays usage guidelines for the statement, while the help reference command displays summary information about the statement.

You can display help based on a system log tag using the help syslog command:

help syslog *syslog-tag*

The help syslog command displays the contents of a syslog message.

Example: Getting Help Based on a String in a Statement Name

Get help about statements that contain the string “traps”:

```
[edit]
user@host# help apropos traps
set interfaces <interface_name>
   Enable SNMP notifications on state changes
set interfaces <interface_name> unit <interface_unit_number>
   Enable SNMP notifications on state changes
set snmp trap-group
   Configure traps and notifications
set snmp trap-group <group_name> version <version> all
   Send SNMPv1 and SNMPv2 traps
set snmp trap-group <group_name> version <version> v1
   Send SNMPv1 traps
set snmp trap-group <group_name> version <version> v2
   Send SNMPv2 traps
set protocols mpls log-updown
   Send SNMP traps
set firewall filter <filter-name> term <rule-name> from source-port snmptrap
   SNMP traps
set firewall filter <filter-name> term <rule-name> from source-port-except snmptrap
   SNMP traps
set firewall filter <filter-name> term <rule-name> from destination-port snmptrap
   SNMP traps
set firewall filter <filter-name> term <rule-name> from destination-port-except
snmptrap
   SNMP traps
set firewall filter <filter-name> term <rule-name> from port snmptrap
   SNMP traps
set firewall filter <filter-name> term <rule-name> from port-except snmptrap
   SNMP traps
[edit]
user@host# edit interfaces at-5/3/0
[edit interfaces at-5/3/0]
user@host# help apropos traps
set <interface_name>
   Enable SNMP notifications on state changes
set <interface_name> unit <interface_unit_number>
   Enable SNMP notifications on state changes
```

Help for Omitted Statements

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

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

Using CLI Command Completion

You do not always have to remember or type the full command or option name for the CLI to recognize it.

- To display all possible command or option completions, type the partial command followed immediately by a question mark.
- To complete a command or option that you have partially typed, press the tab key or the spacebar. If the partially typed letters begin a string that uniquely identifies a command, the complete command name appears. Otherwise, a beep indicates that you have entered an ambiguous command, and the possible completions are displayed.

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

Examples: Using CLI Command Completion

Issue the show interfaces command:

```
user@host> sh<Space>ow i<Space>
'i' is ambiguous.
Possible completions:
     igmp      Show information about IGMP
     interfaceShow interface information
     isis      Show information about IS-IS
user@host> show in<Space>terfaces <Enter>
Physical interface: at-0/1/0, Enabled, Physical link is Up
  Interface index: 11, SNMP ifIndex: 65
  Link-level type: ATM-PVC, MTU: 4482, Clocking: Internal, SONET mode
  Speed: OC12, Loopback: None, Payload scrambler: Enabled
  Device flags   : Present Running
  Link flags     : 0x01
...
user@host>
```

Display a list of all log files whose names start with the string "messages," and then display the contents of one of the files:

```
user@myhost> show log mes?
Possible completions:
  <filename>          Log file to display
  messages            Size: 1417052, Last changed: Mar  3 00:33
  messages.0.gz       Size: 145575, Last changed: Mar  3 00:00
  messages.1.gz       Size: 134253, Last changed: Mar  2 23:00
  messages.10.gz      Size: 137022, Last changed: Mar  2 14:00
  messages.2.gr       Size: 137112, Last changed: Mar  2 22:00
  messages.3.gz       Size: 121633, Last changed: Mar  2 21:00
  messages.4.gz       Size: 135715, Last changed: Mar  2 20:00
  messages.5.gz       Size: 137504, Last changed: Mar  2 19:00
  messages.6.gz       Size: 134591, Last changed: Mar  2 18:00
  messages.7.gz       Size: 132670, Last changed: Mar  2 17:00
  messages.8.gz       Size: 136596, Last changed: Mar  2 16:00
  messages.9.gz       Size: 136210, Last changed: Mar  2 15:00
user@myhost> show log mes<Tab>sages.4<Tab>.gz<Enter>
Jan 15 21:00:00 myhost newsyslog[1381]: logfile turned over
...
```

Using Command Completion in Configuration Mode

The CLI command completion functions also apply to the commands in configuration mode and to configuration statements. Specifically, to display all possible commands or statements, type the partial string followed immediately by a question mark; to complete a command or statement that you have partially typed, press the tab key or spacebar.

Command completion also applies to identifiers, with one slight difference. To display all possible identifiers, type a partial string followed immediately by a question mark. To complete an identifier, you must press the tab key. This scheme allows you to enter identifiers with similar names; then press the spacebar when you are done typing the identifier name.

Examples: Using Command Completion in Configuration Mode

List the configuration mode commands:

```
user@host# ?
Possible completions:
<[Enter]>        Execute this command
  activate       Remove the inactive tag from a statement
  annotate       Annotate the statement with a comment
  commit         Commit current set of changes
  copy           Copy a statement
  deactivate     Add the inactive tag to a statement
  delete         Delete a data element
  edit           Edit a sub-element
  exit           Exit from this level
  help           Provide help information
  insert         Insert a new ordered data element
  load           Load configuration from an ASCII file
  quit           Quit from this level
  rename         Rename a statement
  rollback       Roll back database to last committed version
  run            Run an operational-mode command
  save           Save configuration to an ASCII file
  set            Set a parameter
  show           Show a parameter
  status         Display database user status
  top            Exit to top level of configuration
  up             Exit one level of configuration
```

List all the statements available at a particular hierarchy level:

```
[edit]
user@host# edit ?
Possible completions:
> accounting-options Accounting data configuration
> chassis            Chassis configuration
> class-of-service   Class-of-service configuration
> firewall           Define a firewall configuration
> forwarding-options Configure options to control packet sampling
> groups             Configuration groups
> interfaces         Interface configuration
> policy-options     Routing policy option configuration
```

```
> protocols             Routing protocol configuration
> routing-instances     Routing instance configuration
> routing-options       Protocol-independent routing option configuration
> snmp                  Simple Network Management Protocol
> system                System parameters

user@host# edit protocols ?
Possible completions:
<[Enter]>           Execute this command
> bgp               BGP options
> connections       Circuit cross-connect configuration
> dvmrp             DVMRP options
> igmp              IGMP options
> isis              IS-IS options
> ldp               LDP options
> mpls              Multiprotocol Label Switching options
> msdp              MSDP options
> ospf              OSPF configuration
> pim               PIM options
> rip               RIP options
> router-discovery  ICMP router discovery options
> rsvp              RSVP options
> sap               Session Advertisement Protocol options
> vrrp              VRRP options
  |                 Pipe through a command
[edit]
user@host# edit protocols
```

List all commands that start with a particular letter or string:

```
user@host# edit routing-options a?
Possible completions:
     > aggregate          Coalesced routes
     > autonomous-systemAutonomous system number
[edit]
user@host# edit routing-options a
```

List all configured Asynchronous Transfer Mode (ATM) interfaces:

```
user@host# edit interfaces at?
Possible completions:
     <interface_name>     Interface name
     at-2/1/1
     at-2/2/0
     at-5/1/0
[edit]
user@host# edit interfaces at
```

Display a list of all configured policy statements:

```
[edit]
user@host# show policy-options policy-statement ?
Possible completions:
<policy_name>        Name to identify a policy filter
[edit]
user@host# edit policy-options policy-statement
```

Displaying Tips About CLI Commands

To get tips about CLI commands, issue the help tip cli command. Each time you enter the command, a new tip appears. For example:

```
user@host> help tip cli
JUNOS tip:
Use 'request system software validate' to validate the incoming software
against the current configuration without impacting the running system.

user@host> help tip cli
JUNOS tip:
Use 'commit and-quit' to exit configuration mode after the commit has
succeeded.  If the commit fails you are left in configuration mode.
```

You can also enter help tip cli <*number*> to associate a tip with a number. This enables you to recall the tip at a later time. For example:

```
user@host> help tip cli 10
JUNOS tip:
Use the re0 and re1 configuration groups to restrict configuration to
a particular routing engine.

user@host> help tip cli
JUNOS tip:
Use the 'save' CLI pipe to save output to a local or remote file.

user@host> help tip cli 10
JUNOS tip:
Use the re0 and re1 configuration groups to restrict configuration to
a particular routing engine.
```

Part 2

Operational Mode and Configuration Mode

Chapter 5

Using CLI Operational Commands to Monitor the Router

This chapter provides information on CLI operational commands.

Topics include:

Overview of CLI Operational Mode Commands

This section provides an overview of operational mode commands.

Topics include:

CLI Command Categories

When you log in to the router and the command-line interface (CLI) starts, you are at the top level of operational mode. At this level, there are a number of broad groups of CLI commands:

- Commands for controlling the CLI environment—The commands in the **set** hierarchy configure the CLI display screen. For information about these commands, see "Controlling the CLI Environment" on page 137.
- Commands for monitoring and troubleshooting—The following commands let you display information and statistics about the software and test network connectivity. Using these commands is discussed in the *JUNOS Interfaces Command Reference*.
  - **clear**—Clear statistics and protocol database information.
  - **mtrace**—Trace mtrace packets from source to receiver.
  - **monitor**—Perform real-time debugging of various software components, including the routing protocols and interfaces.
  - **ping**—Determine the reachability of a remote network host.
  - **show**—Display the current configuration and information about interfaces, routing protocols, routing tables, routing policy filters, system alarms, and the chassis.
  - **test**—Test the configuration and application of policy filters and autonomous system (AS) path regular expressions.
  - **traceroute**—Trace the route to a remote network host.
- Commands for connecting to other network systems—The **ssh** command opens secure shell connections, and the **telnet** command opens telnet sessions to other hosts on the network. For information about these commands, see the *JUNOS System Basics and Services Command Reference*.

- Commands for copying files—The **copy** command copies files from one location on the router to another, from the router to a remote system, or from a remote system to the router. For information about these commands, see the *JUNOS System Basics and Services Command Reference*.

- Commands for restarting software processes—The commands in the **restart** hierarchy restart the various JUNOS software processes, including the routing protocol, interface, and Simple Network Management Protocol (SNMP). For information about these commands, see the *JUNOS System Basics and Services Command Reference*.

- A command—**request**—for performing system-level operations, including stopping and rebooting the router and loading JUNOS software images. For information about this command, see the *JUNOS System Basics and Services Command Reference*.

- A command—**start**—to exit the CLI and start a UNIX shell. For information about this command, see the *JUNOS System Basics and Services Command Reference*.

- A command—**configure**—for entering configuration mode, which provides a series of commands that configure the JUNOS software, including the routing protocols, interfaces, network management, and user access. For information about the CLI configuration commands, see "Using Commands and Statements to Configure the Router" on page 71.

- A command—**quit**—to exit the CLI. For information about this command, see the *JUNOS System Basics and Services Command Reference*.

- For more information about the CLI operational mode commands, see the *JUNOS Interfaces Command Reference* and the *JUNOS System Basics and Services Command Reference*.

Commonly Used Operational Mode Commands

Table 7 lists some operational commands you may find useful for monitoring router operation. For a complete description of operational commands, see the JUNOS command references.

Table 7: Commonly Used Operational Mode Commands

Items to Check	Description	Command
Software version	Versions of software running on the router	show version
Log files	Contents of the log files	monitor
	Log files and their contents and recent user logins	show log
Remote systems	Host reachability and network connectivity	ping
	Route to a network system	traceroute
Configuration	Current system configuration	show configuration
Manipulate files	List of files and directories on the router	file list
	Contents of a file	file show
Interface information	Detailed information about interfaces	show interfaces
Chassis	Chassis alarm status	show chassis alarms
	Information currently on craft display	show chassis craft-interface
	Router environment information	show chassis environment
	Hardware inventory	show chassis hardware
Routing table information	Information about entries in the routing tables	show route
Forwarding table information	Information about data in the kernel's forwarding table	show route forwarding-table
IS-IS	Adjacent routers	show isis adjacency
OSPF	Adjacent routers	clear ospf neighbor
BGP	Entries in the BGP neighbor database	clear bgp neighbor
MPLS	Status of interfaces on which MPLS is running	show mpls interface
	Configured LSPs on the router, as well as all ingress, transit, and egress LSPs	show mpls lsp
	Routes that form a label-switched path	show route label-switched path
RSVP	Status of interfaces on which RSVP is running	show rsvp interface
	Currently active RSVP sessions	show rsvp session
	RSVP packet and error counters	show rsvp statistics

Using Operational Mode Commands

This section describes some general features found in CLI operational mode commands and provides some examples.

Topics include:

Commands with Brief, Detail, Extensive, or Terse Options

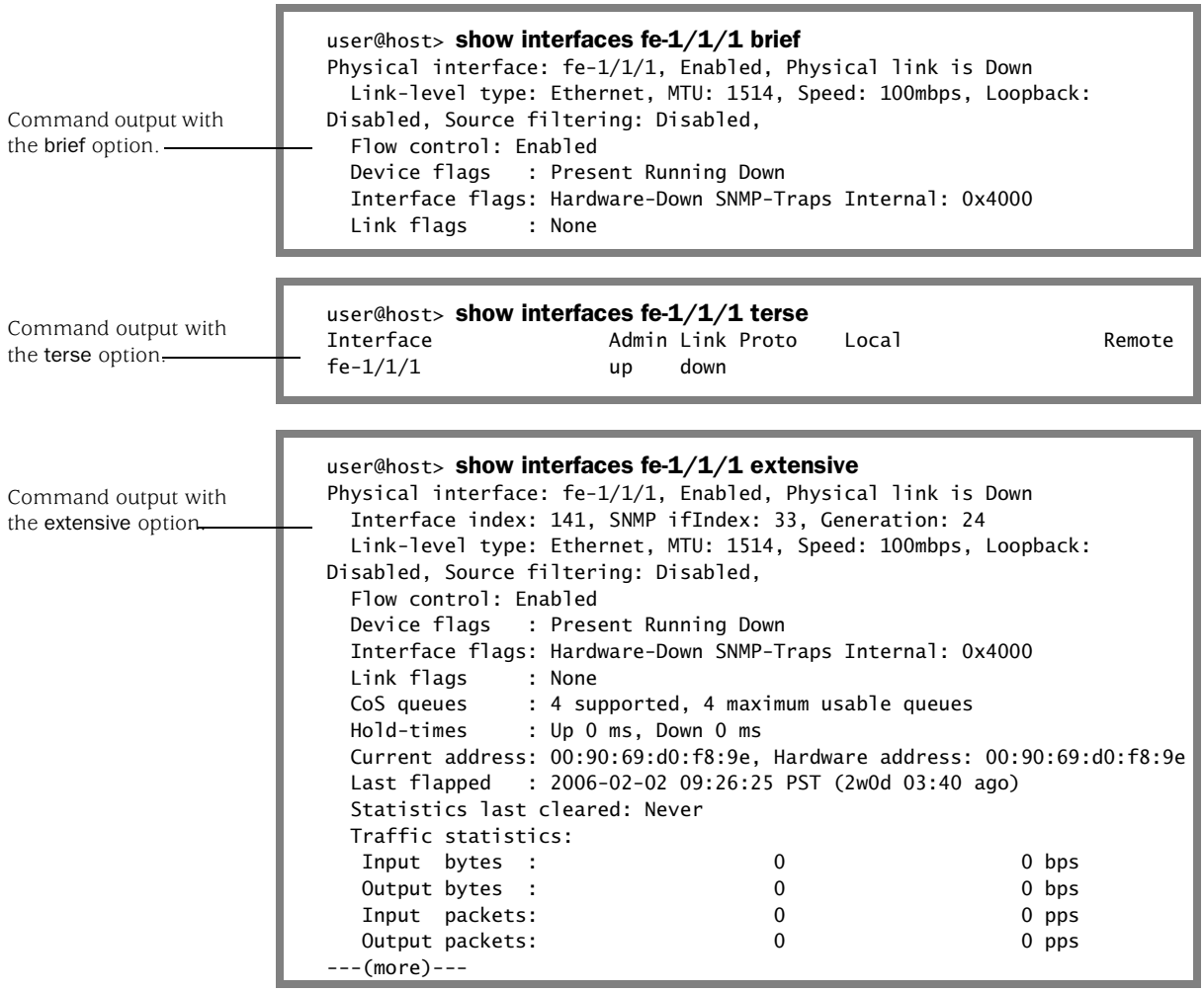
Operational commands can include brief, detail, extensive, or terse options. Use these options to control the amount of information you want to view.

1. Use the ? prompt to list options available for the command. For example:

```
user@host> show interfaces fe-1/1/1 ?
Possible completions:
<[Enter]>            Execute this command
  brief              Display brief output
  descriptions       Display interface description strings
  detail             Display detailed output
  extensive          Display extensive output
  media              Display media information
  snmp-index         SNMP index of interface
  statistics         Display statistics and detailed output
  terse              Display terse output
  |                  Pipe through a command
```

2. Choose the option you wish to use with the command. (See Figure 11.)

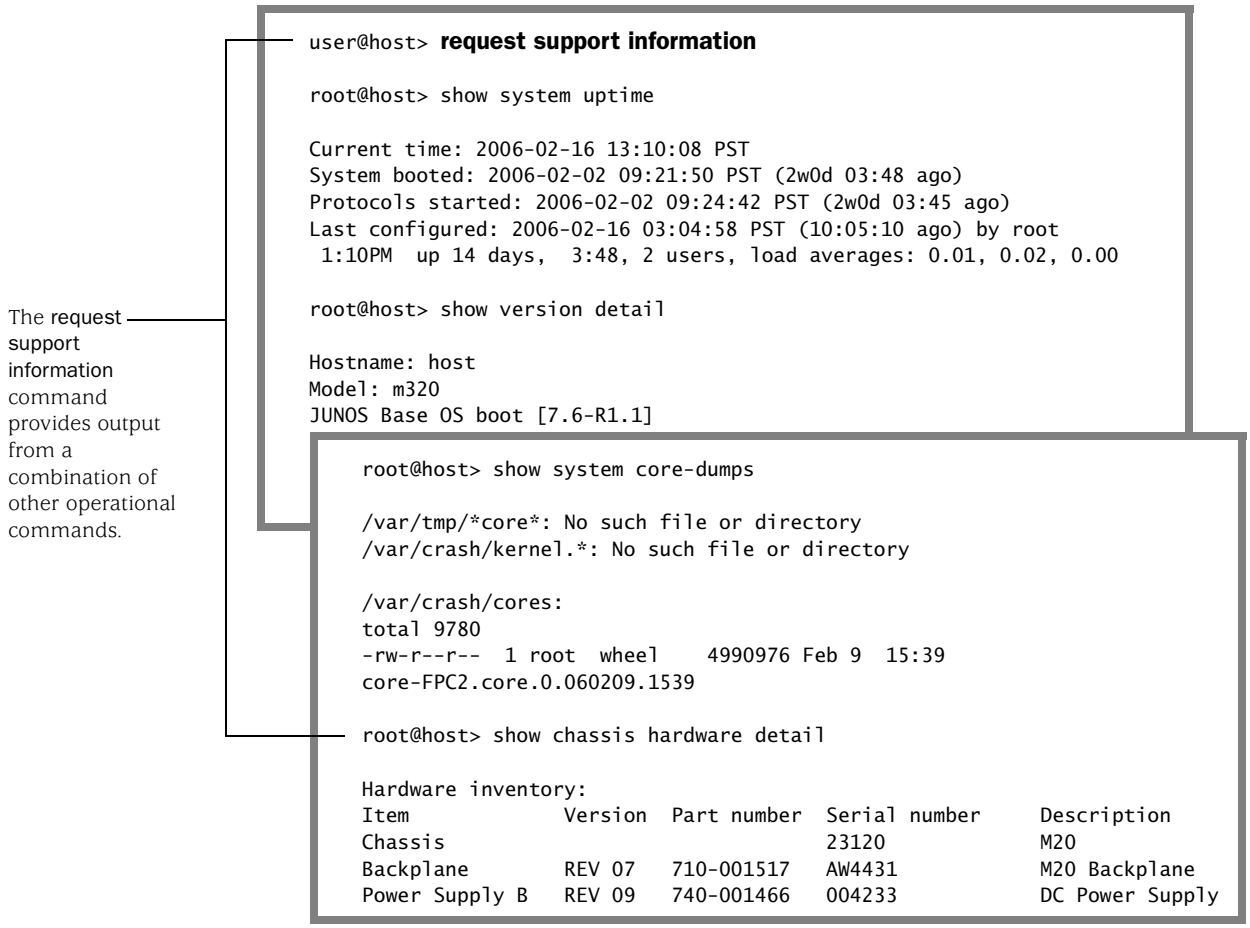
Figure 11: Command Output Options



Commands that Combine Other Commands

In some cases, an operational command is created from a combination of other operational commands. These commands can be useful shortcuts for collecting information about the router. (See Figure 12.)

Figure 12: Commands that Combine Other Commands



Controlling the Scope of a Command

CLI operational commands include options you can use to identify specific components on the router. For example:

1. Type the show interfaces command to display information about all interfaces on the router.

```
user@host> show interfaces
Physical interface: so-0/0/0, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 260
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: OC3,
Loopback: None, FCS: 16,
  Payload scrambler: Enabled
  Device flags   : Present Running Up
  Interface flags: Hardware-Down Point-To-Point SNMP-Traps Internal: 0x4000
  Link flags     : Keepalives
  CoS queues     : 4 supported, 4 maximum usable queues
  Last flapped   : 2006-02-02 09:26:25 PST (2w0d 06:20 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  SONET alarms   : LOL, LOS
  SONET defects  : LOL, LOF, LOS, SEF, AIS-L, AIS-P

Physical interface: so-0/0/1, Enabled, Physical link is Up
  Interface index: 129, SNMP ifIndex: 271
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: OC3,
Loopback: None, FCS: 16,
  Payload scrambler: Enabled
—(more)—
```

2. To display information about a specific interface, type that interface as a command option:

```
user@host> show interfaces fe-1/1/1
Physical interface: fe-1/1/1, Enabled, Physical link is Up
  Interface index: 141, SNMP ifIndex: 33
  Link-level type: Ethernet, MTU: 1514, Speed: 100mbps, Loopback: Disabled,
Source filtering: Disabled,
  Flow control: Enabled
  Device flags   : Present Running Up
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 4 supported, 4 maximum usable queues
  Current address: 00:90:69:d0:f8:9e, Hardware address: 00:90:69:d0:f8:9e
  Last flapped   : 2006-02-02 09:26:25 PST (2w0d 06:25 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : LINK
  Active defects : LINK

user@host>
```

For more information on specifying interface names as command options, see "Using the Comment Character #" on page 68.

Operational Mode Commands on a TX Matrix Platform

When you issue operational mode commands on the TX Matrix platform, CLI command options allow you to restrict the command output to show only a component of the routing matrix rather than to the routing matrix as a whole.

These are the options shown in the CLI:

- scc—The TX Matrix platform.
- lcc *number*—A specific T640 routing node.
- all-lcc—All T640 routing nodes.

If you specify none of these options, then the command applies by default to the whole routing matrix: the TX Matrix platform and all connected T640 routing nodes.

Examples of Routing Matrix Command Options

The following output samples, using the show version command, demonstrate some different options for viewing information about the routing matrix.

```
user@host> show version ?
Possible completions:
  <[Enter]>            Execute this command
  all-lcc              Show software version on all LCC chassis
  brief                Display brief output
  detail               Display detailed output
  lcc                  Show software version on specific LCC (0..3)
  scc                  Show software version on the SCC
  |                    Pipe through a command
```

Sample Output: No Routing Matrix Options Specified

```
user@host> show version
scc-re0:
--------------------------------------------------------------------------
Hostname: scc
Model: TX Matrix
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]

lcc0-re0:
--------------------------------------------------------------------------
Hostname: lcc0
Model: t640
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
JUNOS Support Tools Package [7.0-20040630.0]
```

```
lcc1-re0:
--------------------------------------------------------------------------
Hostname: lcc1
Model: t640
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
JUNOS Support Tools Package [7.0-20040630.0]
```

Sample Output: TX Matrix Platform Only (scc Option)

```
user@host> show version scc
Hostname: scc
Model: TX Matrix
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
```

Sample Output: Specific T640 Routing Node (lcc *number* Option)

```
user@host> show version lcc 0
lcc0-re0:
--------------------------------------------------------------------------
Hostname: lcc0
Model: t640
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
JUNOS Support Tools Package [7.0-20040630.0]
```

Sample Output: All T640 routing nodes (all-lcc Option)

```
user@host> show version all-lcc
lcc0-re0:
--------------------------------------------------------------------------
Hostname: lcc0
Model: t640
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
JUNOS Support Tools Package [7.0-20040630.0]
```

```
lcc1-re0:
--------------------------------------------------------------------------
Hostname: lcc1
Model: t640
JUNOS Base OS boot [7.0-20040630.0]
JUNOS Base OS Software Suite [7.0-20040629.0]
JUNOS Kernel Software Suite [7.0-20040630.0]
JUNOS Packet Forwarding Engine Support (T-Series) [7.0-20040630.0]
JUNOS Routing Software Suite [7.0-20040630.0]
JUNOS Online Documentation [7.0-20040630.0]
JUNOS Crypto Software Suite [7.0-20040630.0]
JUNOS Support Tools Package [7.0-20040630.0]
```

Monitoring Who Uses the CLI

Depending upon how you configure the JUNOS software, multiple users can log in to the router, use the CLI, and configure or modify the software configuration.

If, when you enter configuration mode, another user is also in configuration mode, a notification message is displayed that indicates who the user is and what portion of the configuration they are viewing or editing:

```
user@host> configure
Entering configuration mode
Current configuration users:
      root terminal p3 (pid 1088) on since 1999-05-13 01:03:27 EDT
        [edit interfaces so-3/0/0 unit 0 family inet]
The configuration has been changed but not committed
```

Interface Naming Conventions

This section discusses interface naming conventions used in operational commands.

Topics include:

Physical Part of an Interface Name

The M-series and T-series routing platforms use one convention, while the J-series routing platform uses another.

- M-series and T-series interface names—On the M-series and T-series platforms, when you display information about an interface, you specify the interface type, the slot in which the Flexible PIC Concentrator (FPC) is installed, the slot on the FPC in which the Physical Interface Card (PIC) is located, and the configured port number.

  In the physical part of the interface name, a hyphen (-) separates the media type from the FPC number, and a slash (/) separates the FPC, PIC, and port numbers:

  type-fpc/pic/port

NOTE: Exceptions to the *type-fpc/pic/port* physical description include the aggregated Ethernet and aggregated SONET/SDH interfaces, which use the syntax ae*number* and as*number*, respectively.

- J-series interface names—On the J-series routing platform, when you display information about an interface, you specify the interface type, the slot in which the Physical Interface Module (PIM) is installed, 0, and the configured port number.

  In the physical part of the interface name, a hyphen (-) separates the media type from the PIM number, and a slash (/) separates the PIM, 0, and port numbers:

  type-pim/0/port

NOTE: An exception to the *type-pim/0/port* physical description is the Integrated Services Digital Network (ISDN) dialer interface, which uses the syntax dl*number*.

Logical Part of an Interface Name

The logical unit part of the interface name corresponds to the logical unit number, which can be a number from 0 through 16,384. In the virtual part of the name, a period (.) separates the port and logical unit numbers:

- M-series and T-series routing platforms:

  type-fpc/pic/port.logical

- J-series routing platform:

  type-pim/0/port.logical

Channel Identifier Part of an Interface Name

The channel identifier part of the interface name is required only on channelized interfaces. For channelized interfaces, channel 0 identifies the first channelized interface. For channelized intelligent queuing (IQ) interfaces, channel 1 identifies the first channelized interface.

NOTE: Depending on the type of channelized interface, up to three levels of channelization can be specified. For more information, see the *JUNOS Network Interfaces Configuration Guide*.

A colon (:) separates the physical and virtual parts of the interface name:

- M-series and T-series routing platforms:

  type-fpc/pic/port:channel
  type-fpc/pic/port:channel:channel
  type-fpc/pic/port:channel:channel:channel

- J-series routing platforms:

  type-pim/0/port:channel
  type-pim/0/port:channel:channel
  type-pim/0/port:channel:channel:channel

Viewing Files and Directories

The JUNOS software stores information in files on the router, including configuration files, log files, and router software files. This section shows some examples of operational commands you can use to view files and directories on the router.

Topics include:

Directories on the Router

Some important directories on the router are shown in Table 8.

Table 8: Directories on the Router

Directory	Description
/config	This directory is located on the router's internal flash drive. It contains the active configuration (juniper.conf) and rollback files 1, 2, and 3.
/var/db/config	This directory is located on the router's hard drive and contains rollback files 4 through 9.
/var/tmp	This directory is located on the router's hard drive. It holds core files from the various daemons on the Routing Engines. Core files are generated when a particular daemon crashes and are used by Juniper Networks engineers to diagnose the reason for failure.
/var/log	This directory is located on the router's hard drive. It contains files generated by both the router's logging function as well as the traceoptions command.
/var/home	This directory is located on the router's hard drive. It contains a subdirectory for each configured user on the router. These individual user directories are the default file location for many JUNOS software commands.
/altroot	This directory is located on the router's hard drive and contains a copy of the root file structure from the internal flash drive. This directory is used in certain disaster-recovery modes where the internal flash drive is not operational.
/altconfig	This directory is located on the router's hard drive and contains a copy of the /config file structure from the internal flash drive. This directory is also used in certain disaster recovery modes where the internal flash drive is not operational.

Listing Files and Directories

You can view the router's directory structure as well as individual files by issuing the file command in operational mode.

1. To get help about the file command, type the following:

```
user@host> file ?
Possible completions:
compare   Compare files (local)
copy      Copy files (local or remote)
delete    Delete files from the system (local)
list      List file information (local)
rename    Rename files (local)
show      Display file contents (local)
```

   Help shows that the file command includes several options for manipulating files.

2. Use the list option to see the directory structure of the router. For example, to show the files located in your home directory on the router:

```
user@host> file list
.ssh/
common
```

The default directory for the file list command is the home directory of the user logged in to the router. In fact, the user's home directory is the default directory for most of the JUNOS software commands requiring a filename.

3. To view the contents of other file directories, specify the directory location. For example:

```
user@host> file list /config
juniper.conf
juniper.conf.1.gz
juniper.conf.2.gz
juniper.conf.3.gz
```

4. You can also use the router's context-sensitive help system to locate a directory. For example:

```
user@host> file list /?
Possible completions:
<[Enter]>       Execute this command
<path>          Path to list
/COPYRIGHT      Size: 4735, Last changed: Mar 31 2001
/altconfig/     Last changed: Dec 11 2001
/altroot/       Last changed: Dec 11 2001
/bin/           Last changed: Aug 26 08:49:25
/boot/          Last changed: Oct 03 16:27:55
/config/        Last changed: Oct 03 16:27:56
/dev/           Last changed: Sep 30 12:10:56
/etc/           Last changed: Oct 03 16:27:56
/kernel         Size: 9302545, Last changed: Apr 30 02:00:21
/mnt/         Last changed: Dec 11 2001
/modules/     Last changed: Aug 26 08:43:17
/packages/    Last changed: Aug 26 08:49:45
/proc/        Last changed: Oct 04 10:20:32
/root/        Last changed: Aug 26 08:47:33
/sbin/        Last changed: Aug 26 08:49:45
/tmp/         Last changed: Oct 03 16:27:55
/usr/         Last changed: Dec 11 2001
/var/         Last changed: Dec 27 2001

user@host> file list /var/?
Possible completions:
<[Enter]>     Execute this command
<path>        Path to list
/var/crash/   Last changed: Sep 16 09:03:30
/var/cron/    Last changed: Dec 27 2001
/var/db/      Last changed: Oct 03 16:27:56
/var/etc/     Last changed: Oct 03 16:27:56
/var/home/    Last changed: Oct 03 15:07:40
/var/log/     Last changed: Oct 03 16:27:56
/var/run/     Last changed: Oct 04 10:07:53
```

```
/var/sw/     Last changed: Dec 27 2001
/var/tmp/    Last changed: Sep 30 12:11:28
```

5. You can also display the contents of a file. For example:

```
user@host> file show /var/log/inventory
Jun 20 20:02:35 CHASSISD release 7.1I0 built by builder on 2005-04-29
07:59:59 UTC
Jun 20 20:02:57 CHASSISD release 7.1I0 built by builder on 2005-04-29
07:59:59 UTC
Jun 20 20:02:59 Routing Engine 0 - part number 740-003239, serial number
9001017848
Jun 20 20:04:00 SSB 0 - part number 710-001951, serial number AG1024
Jun 20 20:04:00 SSRAM bank 0 - part number 710-001385, serial number
214095
Jun 20 20:04:00 SSRAM bank 1 - part number 710-001385, serial number
214241
Jun 20 20:04:00 SSRAM bank 2 - part number 710-001385, serial number
214192
...
```

Specifying Filenames and URLs

In some command-line interface (CLI) commands and configuration statements—including file copy, file archive, load, save, set system login user *username* authentication *load-key-file*, and request system software add—you can include a filename. On a routing matrix, you can include chassis information as part of the filename (for example, lcc0, lcc0-re0, or lcc0-re1).

You can specify a filename or URL in one of the following ways:

- *filename*—File in the user's current directory on the local flash drive. You can use wildcards to specify multiple source files or a single destination file. Wildcards are not supported in Hypertext Transfer Protocol (HTTP) or FTP.

NOTE: Wildcards are supported only by the file (compare | copy | delete | list | rename | show) commands. When you issue the file show command with a wildcard, it must resolve to one filename.

- */path/filename*—File on the local flash disk.
- /var/*filename* or /var/*path*/*filename*—File on the local hard disk. You can also specify a file on a local Routing Engine for a specific T640 routing node on a routing matrix:

```
user@host> file delete lcc0-re0:/var/tmp/junk
```

- a:*filename* or a:*path/filename*—File on the local drive. The default path is / (the root-level directory). The removable media can be in MS-DOS or UNIX (UFS) format.

- *hostname:/path/filename, hostname:filename, hostname:path/filename,* or scp://*hostname/path/filename*—File on an scp/ssh client. This form is not available in the worldwide version of the JUNOS software. The default path is the user's home directory on the remote system. You can also specify *hostname* as *username@hostname*.

- ftp://*hostname/path/filename*—File on an FTP server. You can also specify *hostname* as *username@hostname* or *username*:*password@hostname*. The default path is the user's home directory. To specify an absolute path, the path must start with %2F; for example, ftp://*hostname*/%2F*path/filename*. To have the system prompt you for the password, specify prompt in place of the password. If a password is required, and you do not specify the password or prompt, an error message is displayed:

```
user@host> file copy ftp://username@ftp.hostname.net//filename
file copy ftp.hostname.net: Not logged in.
user@host> file copy ftp://username:prompt@ftp.hostname.net//filename
Password for username@ftp.hostname.net:
```

- http://*hostname*/*path*/*filename*—File on an HTTP server. You can also specify *hostname* as *username@hostname* or *username*:*password@hostname*. If a password is required and you omit it, you are prompted for it.

- re0:/*path/filename* or re1:/*path*/*filename*—File on a local Routing Engine.You can also specify a file on a local Routing Engine for a specific T640 routing node on a routing matrix:

```
user@host> show log lcc0-re1:chassisd
```

Managing Programs and Processes

This section shows some examples of operational commands you can use to manage programs and processes on the router.

Topics include:

Displaying JUNOS Software Information

You can display JUNOS software version information and other status to determine if the version of JUNOS software that you are running supports particular features or hardware.

To display JUNOS software information:

1. Make sure you are in operational mode.
2. To display brief information and status for the kernel and Packet Forwarding Engine, type the following command:

   user@host> **show version brief**

   This command shows version information for the JUNOS software packages installed on the router. For example:

   ```
   Model: m160
   JUNOS Base OS boot [7.6R1.0]
   JUNOS Base OS Software Suite [7.6R1.0]
   JUNOS Kernel Software Suite [7.6R1.0]
   JUNOS Packet Forwarding Engine Support (M160) [7.6R1.0]
   JUNOS Routing Software Suite [7.6R1.0]
   JUNOS Online Documentation [7.6R1.0]
   JUNOS Crypto Software Suite [7.6R1.0]
   JUNOS Support Tools Package [7.6R1.0]
   ```

   If the JUNOS Crypto Software Suite is listed, the router has Canada and USA encrypted JUNOS software. If the JUNOS Crypto Software Suite is not listed, the router is running worldwide nonencrypted JUNOS software.

3. To display detailed version information, type the following command:

   user@host> **show version detail**

   This command displays shows the hostname and version information for the JUNOS software packages installed on your router. It also includes the version information for each software process. For example:

```
Hostname: my-router.net
Model: m160
JUNOS Base OS boot [7.6R1.0]
JUNOS Base OS Software Suite [7.6R1.0]
JUNOS Kernel Software Suite [7.6R1.0]
JUNOS Packet Forwarding Engine Support (M20/M40) [7.6R1.0]
JUNOS Routing Software Suite [7.6R1.0]
JUNOS Online Documentation [7.6R1.0]
JUNOS Crypto Software Suite [7.6R1.0]
JUNOS Support Tools Package [7.6R1.0]
jpfe-common: JUNOS Packet Forwarding Engine Support (M/T Common) [7.6R1.0]
systestelf.pkg: labpkg [7.0]
KERNEL 7.6R1.0 #0 built by builder on 2005-12-15 22:42:19 UTC
MGD release 7.6R1.0 built by builder on 2005-12-15 22:42:50 UTC
CLI release 7.6R1.0 built by builder on 2005-12-15 22:42:29 UTC
CHASSISD release 7.6R1.0 built by builder on 2005-12-15 22:31:33 UTC
DFWD release 7.6R1.0 built by builder on 2005-12-15 22:33:30 UTC
DCD release 7.6R1.0 built by builder on 2005-12-15 22:30:31 UTC
RPD release 7.6R1.0 built by builder on 2005-12-15 22:43:01 UTC
SNMPD release 7.6R1.0 built by builder on 2005-12-15 22:36:55 UTC
MIB2D release 7.6R1.0 built by builder on 2005-12-15 22:35:10 UTC
APSD release 7.6R1.0 built by builder on 2005-12-15 22:31:27 UTC
VRRPD release 7.6R1.0 built by builder on 2005-12-15 22:37:27 UTC
ALARMD release 7.6R1.0 built by builder on 2005-12-15 22:31:21 UTC
PFED release 7.6R1.0 built by builder on 2005-12-15 22:35:59 UTC
CRAFTD release 7.6R1.0 built by builder on 2005-12-15 22:33:14 UTC
SAMPLED release 7.6R1.0 built by builder on 2005-12-15 22:36:45 UTC
ILMID release 7.6R1.0 built by builder on 2005-12-15 22:34:47 UTC
RMOPD release 7.6R1.0 built by builder on 2005-12-15 22:36:27 UTC
COSD release 7.6R1.0 built by builder on 2005-12-15 22:33:01 UTC
KMD release 7.6R1.0 built by builder on 2005-12-15 22:27:33 UTC
FSAD release 7.6R1.0 built by builder on 2005-12-15 22:34:10 UTC
IRSD release 7.6R1.0 built by builder on 2005-12-15 22:34:51 UTC
NASD release 7.6R1.0 built by builder on 2005-12-15 22:35:53 UTC
FUD release 7.6R1.0 built by builder on 2005-12-15 22:34:14 UTC
PPMD release 7.6R1.0 built by builder on 2005-12-15 22:42:57 UTC
LMPD release 7.6R1.0 built by builder on 2005-12-15 22:42:44 UTC
RTSPD release 7.6R1.0 built by builder on 2005-12-15 22:36:39 UTC
SMARTD release 7.6R1.0 built by builder on 2005-12-15 22:30:06 UTC
KSYNCD release 7.6R1.0 built by builder on 2005-12-15 22:34:59 UTC
LRMUXD release 7.6R1.0 built by builder on 2005-12-15 22:42:47 UTC
SPD release 7.6R1.0 built by builder on 2005-12-15 22:37:12 UTC
ECCD release 7.6R1.0 built by builder on 2005-12-15 22:34:09 UTC
PGMD release 7.6R1.0 built by builder on 2005-12-15 22:42:55 UTC
BFDD release 7.6R1.0 built by builder on 2005-12-15 22:42:26 UTC
L2TPD release 7.6R1.0 built by builder on 2005-12-15 22:37:33 UTC
HTTPD release 7.6R1.0 built by builder on 2005-12-15 22:27:29 UTC
SDXD release 7.6R1.0 built by builder on 2005-12-15 22:28:01 UTC
---(more)---
```

Showing Software Processes

To verify system operation or to begin diagnosing an error condition, you may need to display information about software processes running on the router.

To show software processes:

1. Make sure you are in operational mode.

2. Type the following command:

   user@host> **show system processes extensive**

This command shows the central processing unit (CPU) utilization on the router and lists the processes in order of CPU utilization. For example:

```
last pid: 750; load averages: 0.00, 0.00, 0.00 up 0+00:58:50 18:34:17
52 processes: 1 running, 51 sleeping
Mem: 50M Active, 19M Inact, 38M Wired, 264K Cache, 86M Buf, 642M Free
Swap: 768M Total, 768M Free

PID USERNAME PRI NICE  SIZE    RES STATE  TIME WCPU    CPU COMMAND
546 root     10   0   9096K  1720K nanslp 0:21 0.00% 0.00% chassisd
685 root      2   0   12716K 3840K kqread 0:01 0.00% 0.00% rpd
553 root      2   0   8792K  1544K select 0:01 0.00% 0.00% mib2d
552 root      2   0   8632K  1556K select 0:01 0.00% 0.00% snmpd
563 root      2   0   9316K  1564K select 0:00 0.00% 0.00% kmd
564 root      2   0   7736K   948K select 0:00 0.00% 0.00% fud
131 root     10   0    770M 25568K mfsidl 0:00 0.00% 0.00% newfs
547 root      2   0   7732K   888K select 0:00 0.00% 0.00% alarmd
545 root      2   0  10292K  2268K select 0:00 0.00% 0.00% dcd
550 root      2 -12   1308K   692K select 0:00 0.00% 0.00% ntpd
  1 root     10   0    816K   520K wait   0:00 0.00% 0.00% init
750 root     32   0  21716K   828K RUN    0:00 0.00% 0.00% top
560 root      2   0   8208K  1088K select 0:00 0.00% 0.00% rmopd
561 root      2   0   8188K  1156K select 0:00 0.00% 0.00% cosd
559 root      2   0   1632K   840K select 0:00 0.00% 0.00% ilmid
```

Table 9 lists and describes the output fields included in this example. The fields are listed in alphabetical order.

Table 9: show system processes Command Extensive Output Fields

Field	Description
COMMAND	Command that is running.
CPU	Raw (unweighted) CPU usage. The value of this field is used to sort the processes in the output.
last pid	Last process identifier assigned to the process.
load averages	Three load averages, followed by the current time.
Mem	Information about physical and virtual memory allocation.
NICE	UNIX "nice" value. The nice value allows a process to change it's final scheduling priority.
PID	Process identifier.
PRI	Current kernel scheduling priority of the process. A lower number indicates a higher priority.
processes	Number of existing processes and the number of processes in each state (sleeping, running, starting, zombies, and stopped).
RES	Current amount of resident memory, in KB.
SIZE	Total size of the process (text, data, and stack), in KB.
STATE	Current state of the process (sleep, wait, run, idle, zombi, or stop).
Swap	Information about physical and virtual memory allocation.
USERNAME	Owner of the process.
WCPU	Weighted CPU usage.

Restarting a JUNOS Software Process

To correct an error condition, you may need to restart a software processes running on the router.

To restart a software process:

1. Make sure you are in operational mode.
2. Type the following command:

   user@host> **restart** <*process*>

   Where <process> is the name of the process you want to restart. For example:

   ```
   user@host> restart routing
   Routing protocol daemon started, pid 751
   ```

   This command restarts the routing protocol daemon. When a process restarts, the process identification (PID) is updated. (See Figure 13 on page 66.)

Figure 13: Restarting a Process

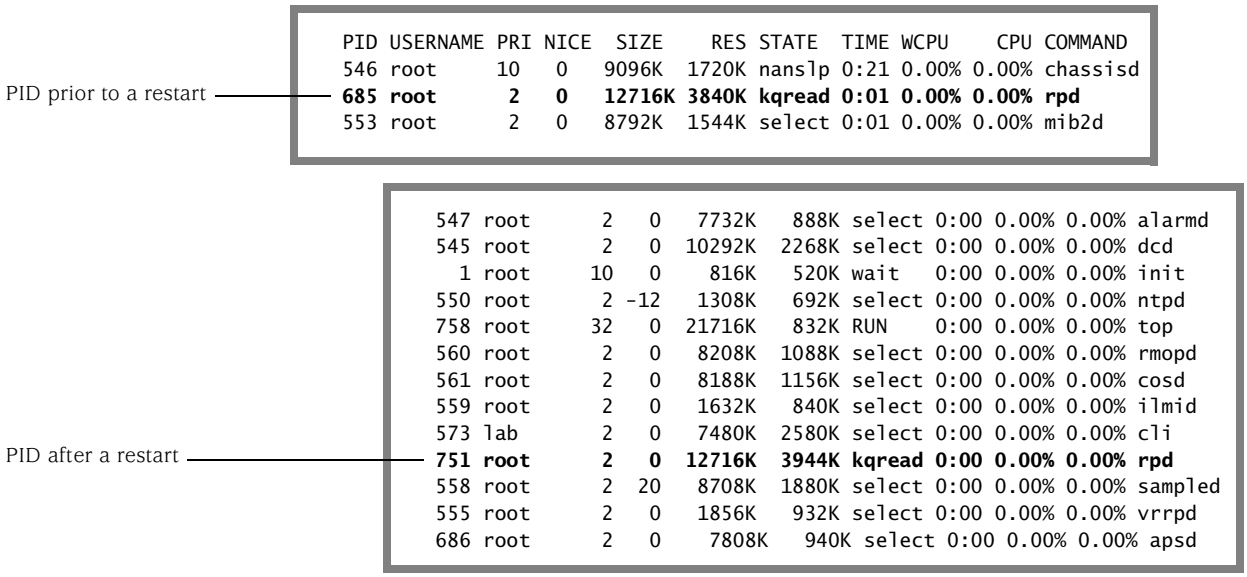


Table 10 shows options available for the restart command.

Table 10: Options to Restart a JUNOS Software Process

Option	Description
class-of-service	Restart the class-of-service process, which controls the router's class-of-service configuration.
gracefully	Restart the software process by sending the equivalent of a UNIX SIGTERM signal.
immediately	Immediately restart the process by sending the equivalent of a UNIX SIGKILL signal.
interface-control	Restart the interface process, which controls the router's physical interface devices and logical interfaces.
mib-process	Restart the Management Information Base (MIB) II process, which provides the router's MIB II agent.
network-access-service	Restart the network access process, which provides the router's Challenge Handshake Authentication Process (CHAP) authentication service.
remote-operations	Restart the remote operations process, which provides the ping and traceroute MIBs.
routing	Restart the routing protocol process, which controls the routing protocols that run on the router and maintains the routing tables.
sampling	Restart the sampling process, which performs packet sampling and cflowd export.

Option	Description
snmp	Restart the Simple Network Management Process (SNMP) process, which provides the router's SNMP master agent.
soft	Reread and reactivate the configuration without completely restarting the software processes. For example, Border Gateway Protocol (BGP) peers stay up and the routing table stays constant. This option is the equivalent of a UNIX SIGHUP signal; omitting this option is the equivalent of a UNIX SIGTERM (kill) operation.

Stopping the JUNOS Software

To avoid damage to the file system, you must gracefully shut down the JUNOS software before powering down the router. If you have configured a backup Routing Engine, it must be shut down before the master Routing Engine.

To stop the JUNOS software:

1. Make sure you are in operational mode.

2. Type the following command:

   user@host> **request system halt**

   This command stops all system processes and halts the operating system. For example:

```
user@host> request system halt
Halt the system ? [yes,no] (no) yes
shutdown: [pid 3110]
Shutdown NOW!

*** FINAL System shutdown message from root@host ***
System going down IMMEDIATELY

user@host> Dec 17 17:28:40 init: syslogd (PID 2514) exited with status=0
Normal Exit
Waiting (max 60 seconds) for system process `bufdaemon' to stop...stopped
Waiting (max 60 seconds) for system process `syncer' to stop...stopped
syncing disks... 4
done
Uptime: 3h31m41s
ata0: resetting devices .. done
The operating system has halted.
Please press any key to reboot.
```

Rebooting the JUNOS Software

After a software upgrade or to recover (occasionally) from an error condition, you will need to reboot the JUNOS software.

To reboot the JUNOS software:

1. Make sure you are in operational mode.
2. Type the following command:

   user@host> **request system reboot**

   This command displays the final stages of the system shutdown and executes the reboot. Reboot requests are recorded to the system log files, which you can view with the show log messages command. For example:

```
Reboot the system ? [yes,no] (no) yes

shutdown: [pid 845]
Shutdown NOW!

*** FINAL System shutdown message from root@host ***
System going down IMMEDIATELY


user@host> Dec 17 17:34:20 init: syslogd (PID 409) exited with status=0
Normal Exit
Waiting (max 60 seconds) for system process `bufdaemon' to stop...stopped
Waiting (max 60 seconds) for system process `syncer' to stop...stopped
syncing disks... 10 6
done
Uptime: 2m45s
ata0: resetting devices .. done
Rebooting...
```

Using the Comment Character

You can copy operational mode commands that include comments from a file and paste them into the CLI. A pound sign (#) at the beginning of the command line indicates a comment line. This is useful for describing frequently used operational mode commands; for example, a user's work instructions on how to monitor the network. To add a comment to a command file, the first character of the line must be #. When you start a command with #, the rest of the line is disregarded by the JUNOS software.

To add comments in operational mode, start with a # and end with a new line (carriage return):

user@host> **#** *comment-string*

comment-string is the text of the comment. The comment text can be any length, and you must type it on a single line.

Example: Using Comments

File with Comments

```
#Command 1: Show the router version
show version
#Command 2: Show all router interfaces
show interfaces terse
```

Copy and Paste Contents of the File into the CLI

```
user@host> #Command 1: Show the router version

user@host> show version
Hostname: myhost
Model: m5
JUNOS Base OS boot [6.4-20040511.0]
JUNOS Base OS Software Suite [6.4-20040511.0]
JUNOS Kernel Software Suite [6.4-20040511.0]
JUNOS Packet Forwarding Engine Support (M5/M10) [6.4-20040511.0] JUNOS
Routing Software Suite [6.4-20040511.0] JUNOS Online Documentation
[6.4-20040511.0] JUNOS Crypto Software Suite [6.4-20040511.0]

user@host> # Command 2: Show all router interfaces
user@host> show interfaces terse
Interface               Admin  Link       Proto Local               Remote
fe-0/0/0                  up   up
fe-0/0/1                  up   down
fe-0/0/2                  up   down
mo-0/1/0                  up
mo-0/1/0.16383            up   up         inet  10.0.0.1            –> 10.0.0.17
so-0/2/0                  up   up
so-0/2/1                  up   up
dsc                       up   up
fxp0                      up   up
fxp0.0                    up   up         inet  192.168.70.62/21
fxp1                      up   up
fxp1.0                    up   up         tnp   4
gre                       up   up
ipip                      up   up
lo0                       up   up
lo0.0                     up   up         inet  127.0.0.1           –> 0/0
lo0.16385                 up   up         inet inet6
```

Chapter 6

Using Commands and Statements to Configure the Router

This chapter describes how to use the CLI to configure the router.

Topics include:

Understanding CLI Configuration Mode

You can configure all properties of the JUNOS software, including interfaces, general routing information, routing protocols, and user access, as well as several system hardware properties.

As described in "Understanding CLI Command Modes" on page 4, a router configuration is stored as a hierarchy of statements. In configuration mode, you create the specific hierarchy of configuration statements that you want to use. When you have finished entering the configuration statements, you commit them, which activates the configuration on the router.

You can create the hierarchy interactively or you can create an ASCII text file that is loaded onto the router and then committed.

Topics in this section include:

Configuration Mode Commands

Table 11 summarizes each CLI configuration mode command. The commands are organized alphabetically.

Table 11: Summary of Configuration Mode Commands (1 of 2)

Command	Description
activate	Remove the inactive: tag from a statement, effectively reading the statement or identifier to the configuration. Statements or identifiers that have been activated take effect when you next issue the commit command.
annotate	Add comments to a configuration. You can add comments only at the current hierarchy level.
commit	Commit the set of changes to the database and cause the changes to take operational effect.
copy	Make a copy of an existing statement in the configuration.
deactivate	Add the inactive: tag to a statement, effectively commenting out the statement or identifier from the configuration. Statements or identifiers marked as inactive do not take effect when you issue the commit command.
delete	Delete a statement or identifier. All subordinate statements and identifiers contained within the specified statement path are deleted with it.
edit	Move inside the specified statement hierarchy. If the statement does not exist, it is created.
exit	Exit the current level of the statement hierarchy, returning to the level prior to the last edit command, or exit from configuration mode. The quit and exit commands are synonyms.
help	Display help about available configuration statements.
insert	Insert an identifier into an existing hierarchy.
load	Load a configuration from an ASCII configuration file or from terminal input. Your current location in the configuration hierarchy is ignored when the load operation occurs.
quit	Exit the current level of the statement hierarchy, returning to the level prior to the last edit command, or exit from configuration mode. The quit and exit commands are synonyms.
rename	Rename an existing configuration statement or identifier.
replace	Replace identifiers or values in a configuration.
rollback	Return to a previously committed configuration. The software saves the last 10 committed configurations, including the rollback number, date, time, and name of the user who issued the commit configuration command.

Table 11: Summary of Configuration Mode Commands (2 of 2)

Command	Description
run	Run a top-level CLI command without exiting from configuration mode.
save	Save the configuration to an ASCII file. The contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it. This allows a section of the configuration to be saved, while fully specifying the statement hierarchy.
set	Create a statement hierarchy and set identifier values. This is similar to edit except that your current level in the hierarchy does not change.
show	Display the current configuration.
status	Display the users currently editing the configuration.
top	Return to the top level of configuration command mode, which is indicated by the [edit] banner.
up	Move up one level in the statement hierarchy.
update	Update a private database.
wildcard	Delete a statement or identifier.

For more information about configuration mode commands, see "Summary of CLI Configuration Mode Commands" on page 193.

Configuration Statements and Identifiers

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

Table 12 describes top-level CLI configuration mode statements.

Table 12: Configuration Mode Top-Level Statements (1 of 2)

Statement	Description
access	Configure the Challenge Handshake Authentication Protocol (CHAP). For information about the statements in this hierarchy, see the *JUNOS System Basics Configuration Guide*.
accounting-options	Configure accounting statistics data collection for interfaces and firewall filters. For information about the statements in this hierarchy, see the *JUNOS Network Management Configuration Guide.*
chassis	Configure properties of the router chassis, including conditions that activate alarms and SONET/SDH framing and concatenation properties. For information about the statements in this hierarchy, see the *JUNOS System Basics Configuration Guide*.
class-of-service	Configure class-of-service parameters. For information about the statements in this hierarchy, see the *JUNOS Class of Service Configuration Guide.*
firewall	Define filters that select packets based on their contents. For information about the statements in this hierarchy, see the *JUNOS Policy Framework Configuration Guide*.

Table 12: Configuration Mode Top-Level Statements (2 of 2)

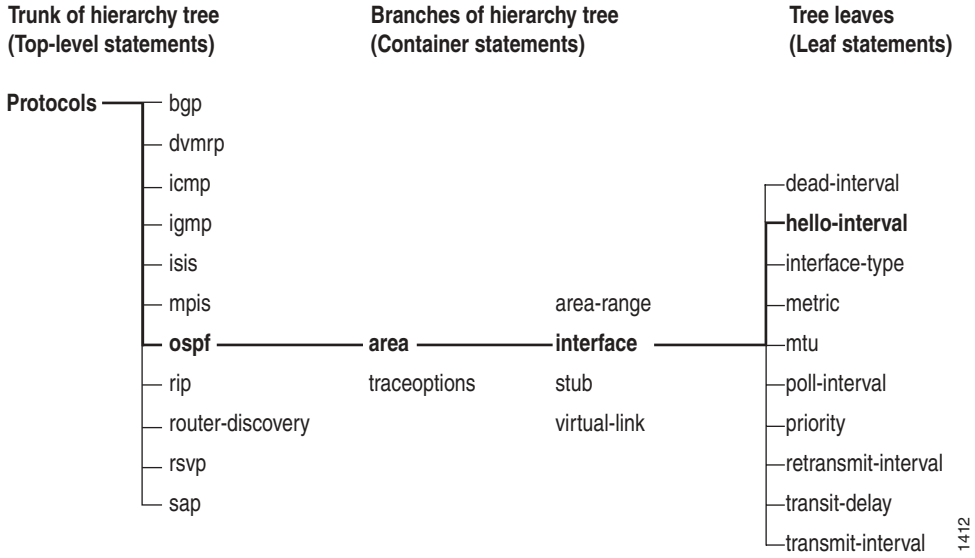
Statement	Description
forwarding-options	Define forwarding options, including traffic sampling options. For information about the statements in this hierarchy, see the *JUNOS Network Interfaces Configuration Guide*.
groups	Configure configuration groups. For information about statements in this hierarchy, see the *JUNOS System Basics Configuration Guide*.
interfaces	Configure interface information, such as encapsulation, interfaces, virtual channel identifiers (VCIs), and data-link connection identifiers (DLCIs). For information about the statements in this hierarchy, see the *JUNOS Network Interfaces Configuration Guide*.
policy-options	Define routing policies, which allow you to filter and set properties in incoming and outgoing routes. For information about the statements in this hierarchy, see the *JUNOS Policy Framework Configuration Guide*.
protocols	Configure routing protocols, including Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), Label Distribution Protocol (LDP), Multiprotocol Label Switching (MPLS), OSPF, Routing Information Protocol (RIP), and Resource Reservation Protocol (RSVP). For information about the statements in this hierarchy, see the chapters that discuss how to configure the individual routing protocols in the *JUNOS Routing Protocols Configuration Guide* and the *JUNOS MPLS Applications Configuration Guide*.
routing-instances	Configure multiple routing instances. For information about the statements in this hierarchy, see the *JUNOS Routing Protocols Configuration Guide*.
routing-options	Configure protocol-independent routing options, such as static routes, autonomous system numbers, confederation members, and global tracing (debugging) operations to log. For information about the statements in this hierarchy, see the *JUNOS Routing Protocols Configuration Guide*.
security	Configure IP Security (IPSec) services. For information about the statements in this hierarchy see the *JUNOS System Basics Configuration Guide*.
snmp	Configure Simple Network Management Protocol (SNMP) community strings, interfaces, traps, and notifications. For information about the statements in this hierarchy, see the *JUNOS Network Management Configuration Guide*.
system	Configure systemwide properties, including the hostname, domain name, Domain Name System (DNS) server, user logins and permissions, mappings between hostnames and addresses, and software processes. For information about the statements in this hierarchy, see the *JUNOS System Basics Configuration Guide*.

For specific information on configuration statements, see the JUNOS configuration guides.

Configuration Statement Hierarchy

The JUNOS software configuration consists of a hierarchy of *statements*. There are two types of statements: *container statements*, which are statements that contain other statements, and *leaf statements*, which do not contain other statements (see Figure 14). All of the container and leaf statements together form the *configuration hierarchy*.

Figure 14: Configuration Mode Hierarchy of Statements



Each statement at the top level of the configuration hierarchy resides at the trunk (or root level) of a hierarchy tree. The top-level statements are container statements, containing other statements that form the tree branches. The leaf statements are the leaves of the hierarchy tree. An individual hierarchy of statements, which starts at the trunk of the hierarchy tree, is called a *statement path*. Figure 14 illustrates the hierarchy tree, showing a statement path for the portion of the protocol configuration hierarchy that configures the hello interval on an interface in an OSPF area.

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

The CLI represents the statement path shown in Figure 14 on page 75 as [protocols ospf area *area-number* interface *interface-name*], and displays the configuration as follows:

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

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

Each leaf statement ends with a semicolon. If the hierarchy does not extend as far as a leaf statement, the last statement in the hierarchy ends with a semicolon.

Entering and Exiting Configuration Mode

You configure the JUNOS software by entering configuration mode and creating a hierarchy of configuration mode statements.

- To enter configuration mode, use the **configure** command.

  When you enter configuration mode, the following configuration mode commands are available:

```
user@host> configure
entering configuration mode
[edit]
user@host# ?
Possible completions:
<[Enter]>       Execute this command
  activate      Remove the inactive tag from a statement
  annotate      Annotate the statement with a comment
  commit        Commit current set of changes
  copy          Copy a statement
  deactivate    Add the inactive tag to a statement
  delete        Delete a data element
  edit          Edit a sub-element
  exit          Exit from this level
  help          Provide help information
  insert        Insert a new ordered data element
  load          Load configuration from an ASCII file
  quit          Quit from this level
  rename        Rename a statement
  rollback      Roll back database to last committed version
```

run	Run an operational-mode command
save	Save configuration to an ASCII file
set	Set a parameter
show	Show a parameter
status	Display database user status
top	Exit to top level of configuration
up	Exit one level of configuration

Users must have configure permission to view and use the configure command. When in configuration mode, a user can view and modify only those statements for which they have access privileges set. For more information, see the *JUNOS System Basics Configuration Guide*.

- If you enter configuration mode and another user is also in configuration mode, a message shows the user's name and what part of the configuration the user is viewing or editing:

```
user@host> configure
Entering configuration mode
Current configuration users:
      root terminal p3 (pid 1088) on since 1999-05-13 01:03:27 EDT
        [edit interfaces so-3/0/0 unit 0 family inet]
The configuration has been changed but not committed
[edit]
user@host>
```

  Up to 32 users can be in configuration mode simultaneously, and they all can be making changes to the configuration. For more information, see "When Multiple Users Configure the Software" on page 98.

- To exit configuration mode, use the exit configuration-mode configuration mode command from any level, or use the exit command from the top level. For example:

```
[edit protocols ospf area 0.0.0.0 interface so-0/0/0]
user@host# exit configuration-mode
exiting configuration mode
user@host>
```

```
[edit]
user@host# exit
exiting configuration mode
user@host>
```

  If you try to exit from configuration mode using the exit command and the configuration contains changes that have not been committed, you see a message and prompt:

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

- To exit with uncommitted changes without having to respond to a prompt, use the exit configuration-mode command. This command is useful when you are using scripts to perform remote configuration.

  ```
  [edit]
  user@host# exit configuration-mode
  The configuration has been changed but not committed
  Exiting configuration mode
  user@host>
  ```

Modifying the Configuration

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

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

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

  edit <*statement-path*>

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

  set <*statement-path*> *statement* <*identifier*>

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

  You cannot use the edit command to change the value of identifiers. You must use the set command.

Topics in this section include:

Displaying the Current Configuration

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

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

The configuration statements appear in a fixed order, and interfaces appear alphabetically by type, and then in numerical order by slot number, PIC number, and port number. Note that when you configure the router, you can enter statements in any order.

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

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

If you have omitted a required statement at a particular hierarchy level, when you issue the show command in configuration mode, a message indicates which statement is missing. As long as a mandatory statement is missing, the CLI continues to display this message each time you issue a show command. For example:

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

Examples: Displaying the Current Configuration

Display the entire configuration:

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/0 hello-interval 5
[edit]
user@host# show
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0 {
                hello-interval 5;
            }
        }
    }
}
```

Display a particular hierarchy in the configuration:

```
[edit]
user@host# show protocols ospf area 0.0.0.0
interface so-0/0/0 {
    hello-interval 5;
}
```

Move down to a level and display the configuration at that level:

```
[edit]
user@host# edit protocols ospf area 0.0.0.0
[edit protocols ospf area 0.0.0.0]
user@host# show
interface so-0/0/0 {
    hello-interval 5;
}
```

Display all of the last committed configuration:

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/0 hello-interval 5
[edit]
user@host# commit
commit complete
[edit]
user@host# quit
exiting configuration mode
user@host> show configuration
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0 {
                hello-interval 5;
            }
        }
    }
}
```

Adding Configuration Statements and Identifiers

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

For example, the following list shows the statements available at the top level of configuration mode.

```
user@host# set ?
Possible completions:
> accounting-options    Accounting data configuration
+ apply-groups          Groups from which to inherit configuration data
> chassis               Chassis configuration
> class-of-service      Class-of-service configuration
> firewall              Define a firewall configuration
> forwarding-options    Configure options to control packet sampling
> groups                Configuration groups
> interfaces            Interface configuration
> policy-options        Routing policy option configuration
> protocols             Routing protocol configuration
> routing-instances     Routing instance configuration
> routing-options       Protocol-independent routing option configuration
> snmp                  Simple Network Management Protocol
> system                System parameters
```

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

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

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

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

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

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

```
[edit]
user@host# set protocols ospf area<Enter>
                                  ^
syntax error, expecting <identifier>.
```

Removing a Statement from the Configuration

To delete a statement or identifier, use the delete configuration mode command. Deleting a statement or an identifier effectively "unconfigures" the functionality associated with that statement or identifier, returning that functionality to its default condition.

```
user@host# delete <statement-path> <identifier>
```

When you delete a statement, the statement and all its subordinate statements and identifiers are removed from the configuration.

For statements that can have more than one identifier, when you delete one identifier, only that identifier is deleted. The other identifiers in the statement remain.

To delete the entire hierarchy starting at the current hierarchy level, do not specify a statement or an identifier in the delete command. When you omit the statement or identifier, you are prompted to confirm the deletion:

```
[edit]
user@host# delete
Delete everything under this level? [yes, no] (no) ?
     Possible completions:
       no      Don't delete everything under this level
       yes     Delete everything under this level
Delete everything under this level? [yes, no] (no)
```

Examples: Removing a Statement from the Configuration

Delete the ospf statement, effectively unconfiguring OSPF on the router:

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface so-0/0/0 hello-interval 5
[edit]
user@host# show
protocols {
     ospf {
        area 0.0.0.0 {
           interface so-0/0/0 {
              hello-interval 5;
           }
        }
     }
}
```

```
[edit]
user@host# delete protocols ospf
[edit]
user@host# show
[edit]
user@host#
```

Delete all statements from the current level down:

```
[edit]
user@host# edit protocols ospf area 0.0.0.0
[edit protocols ospf area 0.0.0.0]
user@host# set interface so-0/0/0 hello-interval 5
[edit protocols ospf area 0.0.0.0]
user@host# delete
Delete everything under this level? [yes, no] (no) yes
[edit protocols ospf area 0.0.0.0]
user@host# show
[edit]
user@host#
```

Unconfigure a particular property:

```
[edit]
user@host# set interfaces so-3/0/0 speed 100mb
[edit]
user@host# show
interfaces {
    so-3/0/0 {
        speed 100mb;
    }
}
[edit]
user@host# delete interfaces so-3/0/0 speed
[edit]
user@host# show
interfaces {
    so-3/0/0;
}
```

For information how to use regular expressions to remove related configuration items, see “Use the opto option to perform a replacement:” on page 149.

Copying a Statement in the Configuration

When you have many statements in a configuration that are similar, you can add one statement, then make copies of that statement. Copying a statement duplicates that statement and the entire hierarchy of statements configured under that statement. Copying statements is useful when you are configuring many physical or logical interfaces of the same type.

To make a copy of an existing statement in the configuration, use the configuration mode copy command:

user@host# **copy** *existing-statement* **to** *new-statement*

Immediately after you have copied a portion of the configuration, the configuration might not be valid. You must check the validity of the new configuration, and if necessary, modify either the copied portion or the original portion for the configuration to be valid.

Example: Copying a Statement in the Configuration

After you have created one virtual connection (VC) on an interface, copy its configuration to create a second VC:

```
[edit interfaces]
user@host# show
at-1/0/0 {
    description "PAIX to MAE West"
    encapsulation atm-pvc;
        unit 61 {
            point-to-point;
            vci 0.61;
            family inet {
                address 10.0.1.1/24;
            }
        }
    }
}
[edit interfaces]
user@host# edit at-1/0/0
[edit interfaces at-1/0/0]
user@host# copy unit 61 to unit 62
[edit interfaces at-1/0/0]
user@host# show
description "PAIX to MAE West"
encapsulation atm-pvc;
    unit 61 {
        point-to-point;
        vci 0.61;
        family inet {
            address 10.0.1.1/24;
        }
    }
```

```
        unit 62 {
            point-to-point;
            vci 0.61;
            family inet {
                address 10.0.1.1/24;
            }
        }
}
```

Issuing Relative Configuration Commands

You can quickly move to the top of the hierarchy or to a level above the area you are configuring. To do this, use the top or up commands followed by another configuration command, including edit, insert, delete, deactivate, annotate, or show.

To issue configuration mode commands from the top of the hierarchy, use the top command; then specify a configuration command. For example:

```
[edit interfaces fxp0 unit 0 family inet]
user@host# top edit system login
[edit system login]
user@host#
```

To issue configuration mode commands from a location higher up in the hierarchy, use the up configuration mode command; specify the number of levels you want to move up the hierarchy; then specify a configuration command. For example:

```
[edit protocols bgp]
user@host# up 2 activate system
```

Renaming an Identifier

When modifying a configuration, you can rename an identifier that is already in the configuration. You can do this either by deleting the identifier (using the delete command) and then adding the renamed identifier (using the set and edit commands), or you can rename the identifier using the rename configuration mode command:

```
user@host# rename <statement-path> identifier1 to identifier2
```

Example: Renaming an Identifier

Change the Network Time Protocol (NTP) server address to 10.0.0.6:

```
[edit]
user@host# rename system network-time server 10.0.0.7 to server 10.0.0.6
```

Inserting a New Identifier

When configuring the router, you can enter most statements and identifiers in any order. Regardless of the order in which you enter the configuration statements, the CLI always displays the configuration in a strict order. However, there are a few cases where the ordering of the statements matters because the configuration statements create a sequence that is analyzed in order.

For example, in a routing policy or firewall filter, you define terms that are analyzed sequentially. Also, when you create a named path in dynamic MPLS, you define an ordered list of the transit routers in the path, starting with the first transit router and ending with the last one.

To modify a portion of the configuration in which the statement order matters, use the insert configuration mode command:

user@host# **insert** *<statement-path> identifier1* **(before | after)** *identifier2*

If you do not use the insert command, but instead simply configure the identifier, it is placed at the end of the list of similar identifiers.

Examples: Inserting a New Identifier

Insert policy terms in a routing policy configuration. Note that if you do not use the insert command, but rather just configure another term, the added term is placed at the end of the existing list of terms.

```
[edit]
user@host# show
policy-options {
    policy-statement statics {
        term term1 {
            from {
                route-filter 192.168.0.0/16 orlonger;
                route-filter 224.0.0.0/3 orlonger;
            }
            then reject;
        }
        term term2 {
            from protocol direct;
            then reject;
        }
        term term3 {
            from protocol static;
            then reject;
        }
        term term4 {
            then accept;
        }
    }
}
[edit]
user@host# rename policy-options policy-statement statics term term4 to term
term6
[edit]
user@host# set policy-options policy-statement statics term term4 from protocol
local
```

```
[edit]
user@host# set policy-options policy-statement statics term term4 then reject
[edit]
user@host# set policy-options policy-statement statics term term5 from protocol
aggregate
[edit]
user@host# set policy-options policy-statement statics term term5 then reject
[edit]
user@host# insert policy-options policy-statement statics term term4 after term
term3
[edit]
user@host# insert policy-options policy-statement statics term term5 after term
term4
[edit]
user@host# show policy-options policy-statement statics
term term1 {
    from {
        route-filter 192.168.0.0/16 orlonger;
        route-filter 224.0.0.0/3 orlonger;
    }
    then reject;
}
term term2 {      # reject direct routes
    from protocol direct;
    then reject;
}
term term3 {       # reject static routes
    from protocol static;
    then accept;
}
term term4 {       #reject local routes
    from protocol local;
    then reject;
}
term term5 {      #reject aggregate routes
    from protocol aggregate;
    then reject;
}
term term6 {   #accept all other routes
    then accept;
}
```

Insert a transit router in a dynamic MPLS path:

```
[edit protocols mpls path ny-sf]
user@host# show
1.1.1.1;
2.2.2.2;
3.3.3.3 loose;
4.4.4.4 strict;
6.6.6.6;
[edit protocols mpls path ny-sf]
user@host# insert 5.5.5.5 before 6.6.6.6
[edit protocols mpls path ny-sf]
user@host# set 5.5.5.5 strict
[edit protocols mpls path ny-sf]
```

```
user@host# show
1.1.1.1;
2.2.2.2;
3.3.3.3 loose;
4.4.4.4 strict;
5.5.5.5 strict;
6.6.6.6;
```

Deactivating and Reactivating Statements and Identifiers

In a configuration, you can deactivate statements and identifiers so that they do not take effect when you issue the commit command. Any deactivated statements and identifiers are marked with the inactive: tag. They remain in the configuration, but are not activated when you issue a commit command.

To deactivate a statement or identifier, use the deactivate configuration mode command:

deactivate (*statement* | *identifier*)

To reactivate a statement or identifier, use the activate configuration mode command:

activate (*statement* | *identifier*)

In both commands, the *statement* or *identifier* you specify must be at the current hierarchy level.

In some portions of the configuration hierarchy, you can include a disable statement to disable functionality. One example is disabling an interface by including the disable statement at the [edit interface *interface-name*] hierarchy level. When you deactivate a statement, that specific object or property is completely ignored and is not applied at all when you issue a commit command. When you disable a functionality, it is activated when you issue a commit command but is treated as though it is down or administratively disabled.

Examples: Deactivating and Reactivating Statements and Identifiers

Deactivate an interface in the configuration:

```
[edit interfaces]
user@host# show
at-5/2/0 {
    traceoptions {
        traceflag all;
    }
    atm-options {
        vpi 0 maximum-vcs 256;
    }
    unit 0 {
...
[edit interfaces]
user@host# deactivate at-5/2/0
[edit interfaces]
user@host# show
```

```
inactive: at-5/2/0 {
    traceoptions {
        traceflag all;
    }
...
```

Reactivate the interface:

```
[edit interfaces]
user@host# activate at-5/2/0
[edit interfaces]
user@host# show
at-5/2/0 {
    traceoptions {
        traceflag all;
    }
...
```

Adding Comments in a Configuration

You can include comments in a configuration to describe any statement in the configuration. You can add comments interactively in the CLI and by editing the ASCII configuration file.

When you add comments in configuration mode, they are associated with a statement at the current level. Each statement can have one single-line comment associated with it. Before you can associate a comment with a statement, the statement must exist. The comment is placed on the line preceding the statement.

To add comments to a configuration, use the annotate configuration mode command:

```
user@host# annotate statement "comment-string"
```

statement is the configuration statement to which you are attaching the comment; it must be at the current hierarchy level. If a comment for the specified *statement* already exists, it is deleted and replaced with the new comment.

comment-string is the text of the comment. The comment text can be any length, and you must type it on a single line. If the comment contains spaces, you must enclose it in quotation marks. In the comment string, you can include the comment delimiters /* */ or #. If you do not specify any, the comment string is enclosed with the /* */ comment delimiters.

To delete an existing comment, specify an empty comment string:

```
user@host# annotate statement ""
```

When you edit the ASCII configuration file and add comments, they can be one or more lines and must precede the statement they are associated with. If you place the comments in other places in the file, such as on the same line following a statement or on a separate line following a statement, they are removed when you use the load command to open the configuration into the CLI.

When you include comments in the configuration file directly, you can format comments in the following ways:

- Start the comment with a /* and end it with a */. The comment text can be on a single line or can span multiple lines.
- Start the comment with a # and end it with a new line (carriage return).

If you add comments with the annotate command, you can view the comments within the configuration by entering the show configuration mode command or the show configuration operational mode command.

When configuring interfaces, you can add comments about the interface by including the description statement at the [edit interfaces *interface-name*] hierarchy level. Any comments you include appear in the output of the show interfaces commands. For more information about the description statement, see the *JUNOS Network Interfaces Configuration Guide*.

Examples: Including Comments in Configurations

Add comments to a configuration:

```
[edit]
user@host# show
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0 {
                hello-interval 5;
            }
        }
    }
}
[edit]
user@host# edit protocols ospf
[edit protocols ospf]
user@host# set area 0.0.0.0
user@host# annotate area 0.0.0.0 "Backbone area configuration added June 15,
1998"
[edit protocols ospf]
user@host# edit area 0.0.0.0
[edit protocols ospf area 0.0.0.0]
user@host# annotate interface so0 "Interface from router sj1 to router sj2"
[edit protocols ospf area 0.0.0.0]
user@host# top
```

```
[edit]
user@host# show
protocols {
    ospf {
        /* Backbone area configuration added June 15, 1998 */
        area 0.0.0.0 {
            /* Interface from router sj1 to router sj2 */
            interface so-0/0/0 {
                hello-interval 5;
            }
        }
    }
}
[edit]
user@host#
```

The following excerpt from a configuration example illustrates how to enter comments in a configuration file:

```
/* This comment goes with routing-options */
routing-options {
    /* This comment goes with routing-options traceoptions */
    traceoptions {
        /* This comment goes with routing-options traceoptions tracefile */
        tracefile rpd size 1m files 10;
        /* This comment goes with routing-options traceoptions traceflag task */
        traceflag task;
        /* This comment goes with routing-options traceoptions traceflag general
*/
        traceflag general;
        }
    autonomous-system 10458;    /* This comment is dropped */
}
routing-options {
    rib-groups {
        ifrg {
            import-rib [ inet.0 inet.2 ];
            /* A comment here is dropped */
        }
        dvmrp-rib {
            import-rib inet.2;
            export-rib inet.2;
            /* A comment here is dropped */
        }
    /* A comment here is dropped */
    }
/* A comment here is dropped */
}
```

Verifying a Configuration

To verify that the syntax of a configuration is correct, use the configuration mode commit check command:

```
[edit]
user@host# commit check
configuration check succeeds
[edit]
user@host#
```

If the commit check command finds an error, a message indicates the location of the error.

Committing a Configuration

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

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

When you enter the commit command, the configuration is first checked for syntax errors (commit check). Then, if the syntax is correct, the configuration is activated and becomes the current, operational router configuration.

You can issue the commit command from any hierarchy level.

These sections discuss how to commit configurations:

If the configuration contains syntax errors, a message indicates the location of the error and the configuration is not activated. The error message has the following format:

```
[edit edit-path]
    'offending-statement;'
      error-message
```

For example:

```
[edit firewall filter login-allowed term allowed from]
    'icmp-type [ echo-request echo-reply ];'
      keyword 'echo-reply' unrecognized
```

You must correct the error before recommitting the configuration. To return quickly to the hierarchy level where the error is located, copy the path from the first line of the error and paste it at the configuration mode prompt at the [edit] hierarchy level.

When you commit a configuration, you commit the entire configuration in its current form. If more than one user is modifying the configuration, committing it saves and activates the changes of all the users.

Committing a Configuration and Exiting Configuration Mode

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

```
[edit]
user@host# commit and-quit
commit complete
exiting configuration mode
user@host>
```

Activating a Configuration but Requiring Confirmation

When you commit the current candidate configuration, you can require an explicit confirmation for the commit to become permanent. This is useful if you want to verify that a configuration change works correctly and does not prevent access to the router. If the change prevents access or causes other errors, the router automatically returns to the previous configuration and restores access after the rollback confirmation timeout passes. This feature is called *automatic rollback*.

To commit the current candidate configuration but require an explicit confirmation for the commit to become permanent, use the commit confirmed configuration mode command:

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

Once you have verified that the change works correctly, you can keep the new configuration active by entering a commit or commit check command within 10 minutes of the commit confirmed command. For example:

```
[edit]
user@host# commit check
commit confirmed will be automatically rolled back in 10 minutes unless confirmed
commit complete
[edit]
user@host#
```

If the commit is not confirmed within a certain amount of time (10 minutes by default), the JUNOS software automatically rolls back to the previous configuration and a broadcast message is sent to all logged-in users.

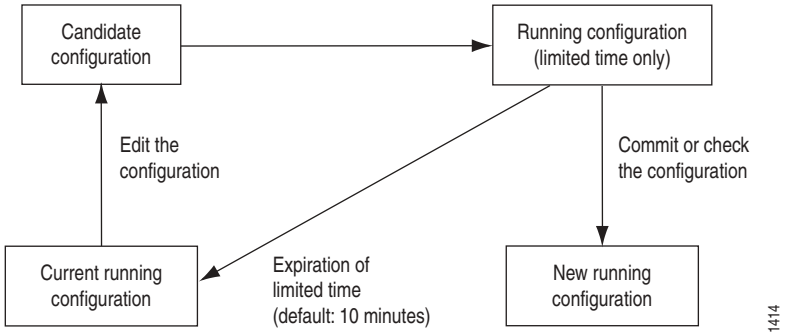
To show when a rollback is scheduled after a commit confirmed command, enter the show system commit command. For example:

```
user@host# show commit confirmed
0 2005-01-05 15:00:37 PST by root via cli commit confirmed, rollback in 3mins
```

Like the commit command, the commit confirmed command verifies the configuration syntax and reports any errors. If there are no errors, the configuration is activated and begins running on the router.

Figure 15 illustrates how the commit confirmed command works.

Figure 15: Confirm a Configuration



To change the amount of time before you have to confirm the new configuration, specify the number of minutes when you issue the command:

```
[edit]
user@host# commit confirmed minutes
commit complete
[edit]
user@host#
```

Scheduling a Commit Operation

You can schedule when you want your candidate configuration to become active. To save software configuration changes and activate the configuration on the router at a future time or upon reboot, use the commit at configuration mode command, specifying reboot or a future time at the [edit] hierarchy level:

```
[edit]
user@host # commit at <string>
```

string is **reboot** or the future time to activate the configuration changes. You can specify time in two formats:

- A time value in the form *hh*:*mm* [:ss] (hours, minutes, and optionally seconds)—Commit the configuration at the specified time, which must be in the future but before 11:59:59 PM on the day the commit at configuration command is issued. Use 24-hour time for the *hh* value; for example, 04:30:00 is 4:30:00 AM, and 20:00 is 8:00 PM. The time is interpreted with respect to the clock and time zone settings on the router.

- A date and time value in the form *yyyy-mm-dd hh*:*mm* [:ss] (year, month, date, hours, minutes, and, optionally, seconds)—Commit the configuration at the specified day and time, which must be after the commit at command is issued. Use 24-hour time for the *hh* value. For example, 2003-08-21 12:30:00 is 12:30 PM on August 21, 2003. The time is interpreted with respect to the clock and time zone settings on the router.

  Enclose the *string* value in quotation marks (“). For example, commit at “18:00:00”. For date and time, include both values in the same set of quotation marks. For example, commit at "2005-03-10 14:00:00".

A “commit check” is performed immediately when you issue the commit at configuration mode command. If the result of the check is successful, then the current user is logged out of configuration mode, and the configuration data is left in a read-only state. No other commit can be performed until the scheduled commit is completed.

NOTE: If the JUNOS software fails before the configuration changes become active, all configuration changes are lost.

You cannot issue the commit at configuration command after you issue the request system reboot command.

You cannot issue the request system reboot command once you schedule a commit operation for a specific time in the future.

You cannot commit a configuration when a scheduled commit is pending. For information about how to cancel a scheduled configuration by means of the clear command, see the *JUNOS System Basics and Services Command Reference*.

Monitoring the Commit Process

To monitor the commit process, use the display detail command after the pipe with the commit command:

```
user@host# commit | display detail
```

For example:

```
[edit]
user@host# commit | display detail
2003-09-22 15:39:39 PDT: exporting juniper.conf
2003-09-22 15:39:39 PDT: setup foreign files
2003-09-22 15:39:39 PDT: propagating foreign files
2003-09-22 15:39:39 PDT: complete foreign files
2003-09-22 15:39:40 PDT: copying configuration to juniper.data+
2003-09-22 15:39:40 PDT: dropping unchanged foreign files
2003-09-22 15:39:40 PDT: daemons checking new configuration
2003-09-22 15:39:41 PDT: commit wrapup...
2003-09-22 15:39:42 PDT: activating '/var/etc/ntp.conf'
2003-09-22 15:39:42 PDT: activating '/var/etc/kmd.conf'
2003-09-22 15:39:42 PDT: activating '/var/db/juniper.data'
2003-09-22 15:39:42 PDT: notifying daemons of new configuration
2003-09-22 15:39:42 PDT: signaling 'Firewall daemon', pid 24567, signal 1,
    status 0
2003-09-22 15:39:42 PDT: signaling 'Interface daemon', pid 24568, signal 1,
    status 0
2003-09-22 15:39:43 PDT: signaling 'Routing protocol daemon', pid 25679,
    signal 1, status 0
2003-09-22 15:39:43 PDT: signaling 'MIB2 daemon', pid 24549, signal 1, status
    0
2003-09-22 15:39:43 PDT: signaling 'NTP daemon', pid 37863, signal 1, status 0
2003-09-22 15:39:43 PDT: signaling 'Sonet APS daemon', pid 24551, signal 1,
    status 0
2003-09-22 15:39:43 PDT: signaling 'VRRP daemon', pid 24552, signal 1, status
    0
2003-09-22 15:39:43 PDT: signaling 'PFE daemon', pid 2316, signal 1, status 0
2003-09-22 15:39:43 PDT: signaling 'Traffic sampling control daemon', pid
24553, signal 1, status 0
2003-09-22 15:39:43 PDT: signaling 'IPSec Key Management daemon', pid
24556, signal 1, status 0
2003-09-22 15:39:43 PDT: signaling 'Forwarding UDP daemon', pid 2320, signal
    1, status 0
commit complete
```

Adding a Comment to Describe the Committed Configuration

You can include a comment that describes changes to the committed configuration. To do so, include the commit comment statement. The comment can be as long as 512 bytes and you must type it on a single line.

```
[edit]
user@host # commit comment <comment-string>
```

comment-string is the text of the comment.

NOTE: You cannot include a comment with the commit check command.

To add a comment to the commit command, include the comment statement after the commit command:

```
[edit]
user@host# commit comment “add user joe”
commit complete
[edit]
user@host#
```

To add a comment to the commit confirmed command, include the comment statement after the commit confirmed command:

```
[edit]
user@host# commit confirmed comment “add customer to port 27”
commit confirmed will be automatically rolled back in 10 minutes unless confirmed
commit complete
[edit]
user@host#
```

To view these commit comments, issue the show system commit operational mode command.

Updating the Alternate Boot Drive

After you commit the configuration and are satisfied that it is running successfully, you should issue the request system snapshot command to back up the new software onto the /altconfig file system. If you do not issue the request system snapshot command, the configuration on the alternate boot drive will be out of sync with the configuration on the primary boot drive.

The request system snapshot command backs up the root file system to /altroot, and /config to /altconfig. The root and /config file systems are on the router's flash drive, and the /altroot and /altconfig file systems are on the router's hard disk (if available).

NOTE: To back up the file system on a J-series Services Router, you must specify a media type (primary compact flash drive, removable compact flash drive, or USB storage device) for backup. For more information, see the *J-series Services Router Administration Guide*.

After you issue the request system snapshot command, you cannot return to the previous version of the software because the running and backup copies of the software are identical.

When Multiple Users Configure the Software

Up to 32 users can be in configuration mode simultaneously, and they all can be making changes to the configuration. All changes made by all users are visible to everyone editing the configuration—the changes become visible as soon as the user presses the Enter key at the end of a command that changes the configuration, such as set, edit, or delete.

When any of the users editing the configuration issues a commit command, all changes made by all users are checked and activated.

Topics in this section include:

Using the Configure Command

If you and other users enter configuration mode with the configure command, everyone can make configuration changes and commit all changes made to the configuration. This means that if you and another user have made configuration changes and the other user commits, the changes you made are committed as well. That is, no one has a lockout on the configuration file.

If, when you enter configuration mode, another user is also in configuration mode, a message shows who the user is and what part of the configuration that user is viewing or editing:

```
user@host> configure
Entering configuration mode
Current configuration users:
     root terminal p3 (pid 1088) on since 1999-05-13 01:03:27 EDT
       [edit interfaces so-3/0/0 unit 0 family inet]
The configuration has been changed but not committed
[edit]
user@host>
```

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

```
user@host> configure
Entering configuration mode
The configuration has been changed but not committed
[edit]
user@host>
```

Displaying Users Currently Editing the Configuration

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

```
user@host# status
Users currently editing the configuration:
  rchen terminal p0 (pid 55691) on since 2006-03-01 13:17:25 PST
     [edit interfaces]
```

The system displays who is editing the configuration (rchen), where the user is logged in (terminal p0), the date and time the user logged in (2006-03-01 13:17:25 PST), and what level of the hierarchy the user is editing ([edit interfaces]).

If you issue the status configuration mode command and a user has scheduled a candidate configuration to become active for a future time, the system displays who scheduled the commit (root), where the user is logged in (terminal d0), the date and time the user logged in (2002-10-31 14:55:15 PST), and that a commit is pending (commit at).

```
[edit]
user@host# status
Users currently editing the configuration:
root terminal d0 (pid 767) on since 2002-10-31 14:55:15 PST, idle 00:03:09
commit at
```

For information about how to schedule a commit, see “Scheduling a Commit Operation” on page 95.

If you issue the status configuration mode command and a user is editing the configuration in configure exclusive mode, the system displays who is editing the configuration (root), where the user is logged in (terminal d0), the date and time the user logged in (2002-11-01 13:05:11 PST), and that a user is editing the configuration in configure exclusive mode (exclusive [edit]).

```
[edit]
user@host# status
Users currently editing the configuration:
root terminal d0 (pid 2088) on since 2002-11-01 13:05:11 PST
exclusive [edit]
```

For more information about configure exclusive, see “Using the Configure Exclusive Command” on page 100.

Using the Configure Exclusive Command

If you enter configuration mode with the configure exclusive command, you lock the candidate *global* configuration (also known as the *shared configuration* or *shared configuration database*) for as long as you remain in configuration mode, allowing you to make changes without interference from other users. Other users can enter and exit configuration mode, but they cannot change the configuration.

If another user has locked the configuration, and you need to forcibly log him or her out, enter the operational mode command request system logout pid *pid_number*.

If you enter configuration mode and another user is also in configuration mode and has locked the configuration, a message indicates who the user is and what portion of the configuration that user is viewing or editing:

```
user@host> configure
Entering configuration mode
Users currently editing the configuration:
    root terminal p3 (pid 1088) on since 2000-10-30 19:47:58 EDT, idle
00:00:44
    exclusive [edit interfaces so-3/0/0 unit 0 family inet]
```

In configure exclusive mode, any uncommitted changes are discarded when you exit:

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

[edit]
user@host# set system host-name cool

[edit]
user@host# quit
The configuration has been changed but not committed
warning: Auto rollback on exiting 'configure exclusive'
Discard uncommitted changes? [yes,no] (yes)

warning: discarding uncommitted changes
load complete
Exiting configuration mode
```

When you use the yes option to exit configure exclusive mode, the JUNOS software discards your uncommitted changes and rolls backs your configuration. The no option allows you to continue editing or to commit your changes in configure exclusive mode.

When a user exits from configure exclusive mode while another user is in configure private mode, the JUNOS software will roll back any uncommitted changes.

Using the Configure Private Command

The configure private command allows multiple users to edit different parts of the configuration at the same time and to commit only their own changes, or to roll back without interfering with one another's changes. When you issue the configure private command, you work in a private candidate configuration, which is a copy of the most recently committed configuration.

NOTE: You cannot enter configure private mode when the global configuration has been modified.

When you commit a private candidate configuration, the JUNOS software temporarily locks the global configuration, enforces the restriction that the global configuration must be unmodified to commit private changes, and validates the private candidate configuration. If a merge conflict occurs, the commit fails and the configuration lock is released. You can then modify your private candidate configuration and commit it again. If there are no errors, the changes made in the private candidate configuration are merged into the most recently committed global configuration, are activated, and begin running on the router, and the configuration lock is released.

NOTE: You cannot commit changes in configure private mode when another user is in configure exclusive mode.

If the global configuration has changed, users in configure private mode can issue the rollback or update command to obtain the most recently committed shared configuration. For more information about the update command, see "Updating the Configure Private Configuration" on page 103.

You must issue the commit command from the top of the configuration.

You cannot save a configure private session; uncommitted changes are discarded.

You cannot issue the commit confirm command when you are in configure private mode.

If a configure private edit is in session, other users who issue the configure command can only view the global configuration; a message appears indicating that these users must use the configure exclusive or configure private commands to modify the configuration:

```
[edit]
user@host# set system host-name ipswitch
error: private edits in use. Try 'configure private' or 'configure
exclusive'.
[edit]
user@host#
```

If the global configuration has been modified, users cannot enter configure private mode because they cannot commit changes when the global configuration has been modified. For example:

```
user@host# configure private
error: shared configuration database modified
Users currently editing the configuration:
root terminal d0 (pid 7951) on since 2002-02-21 14:18:46 PST
[edit]
user@host#
```

NOTE: Users in configure private or configure exclusive mode cannot exit the global configuration with uncommitted changes.

If another user commits a change to the same section of the configuration that the private user has modified, a merge conflict may result. In this case, the JUNOS software updates the private user's configuration with the most recently committed global configuration, then allows the private user can commit the changes. For example:

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

[edit]
user@host# show | compare
[edit system]
-  host-name host;
+  host-name foo;

[edit]
user@host# commit
[edit system host-name]
  'host-name bar'
    statement does not match patch; 'bar' != 'host'
load complete (1 errors)

[edit]
user@host# show | compare
[edit system]
-  host-name bar;
+  host-name foo;

[edit]
user@host#
```

In this example, after the JUNOS software detects the merge conflict and fixes it, the user in configure private mode issues the show | compare command. This command displays the private user's database changes against the most recently committed shared configuration.

Updating the Configure Private Configuration

When you are in configure private mode, you must work with a copy of the most recently committed shared configuration. If the global configuration changes, you can issue the update command to update your private candidate configuration. When you do this, your private candidate configuration contains a copy of the most recently committed configuration with your private changes merged in. For example:

```
[edit]
user@host# update

[edit]
user@host#
```

NOTE: You can get merge conflicts when you issue the update command.

You can also issue the rollback command to discard your private candidate configuration changes and obtain the most recently committed configuration:

```
[edit]
user@host# rollback

[edit]
user@host#
```

Displaying set Commands from the Configuration

In configuration mode, you can display the configuration as a series of configuration mode commands required to recreate the configuration. This is useful if you are not familiar with how to use configuration mode commands or if you want to cut, paste, and edit the displayed configuration. For information about the set command, see “Displaying the Current Configuration” on page 79.

To display the configuration as a series of configuration mode commands required to recreate the configuration from the top level of the hierarchy as set commands, issue the show configuration mode command with the | display set option:

```
user@host# show | display set
```

Example: Displaying set Commands from the Configuration

Display the set commands from the configuration at the [edit interfaces] hierarchy level:

```
[edit interfaces fe-0/0/0]
user@host# show
unit 0 {
    family inet {
        address 192.107.1.230/24;
    }
    family iso;
    family mpls;
}
inactive: unit 1 {
    family inet {
        address 10.0.0.1/8;
    }
}
user@host# show | display set
set interfaces fe-0/0/0 unit 0 family inet address 192.107.1.230/24
set interfaces fe-0/0/0 unit 0 family iso
set interfaces fe-0/0/0 unit 0 family mpls
set interfaces fe-0/0/0 unit 1 family inet address 10.0.0.1/8
deactivate interfaces fe-0/0/0 unit 1
```

To display the configuration as a series of configuration mode commands required to recreate the configuration from the current hierarchy level, issue the show configuration mode command with the | display set relative option:

```
user@host# show | display set relative
```

Example: Displaying Required set Commands at the Current Hierarchy Level

Display the configuration as a series of configuration mode commands required to recreate the configuration from the current hierarchy level:

```
[edit interfaces fe-0/0/0]
user@host# show
unit 0 {
    family inet {
        address 192.107.1.230/24;
    }
    family iso;
    family mpls;
}
inactive: unit 1 {
    family inet {
        address 10.0.0.1/8;
    }
}
user@host# show | display set relative
set unit 0 family inet address 192.107.1.230/24
set unit 0 family iso
set unit 0 family mpls
set unit 1 family inet address 10.0.0.1/8
deactivate unit 1
```

To display the configuration as set commands and search for text matching a regular expression by filtering output, specify the match option after the pipe:

user@host# **show | display set | match** *regular-expression*

Example: Displaying set Commands with the Match Option

Display IP addresses associated with an interface:

```
ge-2/3/0 {
    unit 0 {
        family inet {
            address 192.107.9.106/30;
        }
    }
}
so-5/1/0 {
    unit 0 {
        family inet {
            address 192.107.9.15/32 {
                destination 192.107.9.192;
            }
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 127.0.0.1/32;
        }
    }
}
user@host# show interfaces | display set | match address
set interfaces ge-2/3/0 unit 0 family inet address 192.168.9.106/30
set interfaces so-5/1/0 unit 0 family inet address 192.168.9.15/32 destination
192.168.9.192
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

Displaying Additional Information About the Configuration

In configuration mode only, to display additional information about the configuration, use the display detail command after the pipe in conjunction with a show command. The additional information includes the help string that explains each configuration statement and the permission bits required to add and modify the configuration statement.

user@host# **show** <*hierarchy-level*> **| display detail**

For example:

```
[edit]
user@host# show | display detail
    ##
    ## version: Software version information
    ## require: system
    ##
    version "3.4R1 [tlim]";
    system {
    ##
    ## host-name: Host name for this router
    ## match: ^[[:alnum:]._-]+$
    ## require: system
    ##
    host-name router-name;
    ##
    ## domain-name: Domain name for this router
    ## match: ^[[:alnum:]._-]+$
    ## require: system
    ##
    domain-name isp.net;
    ##
    ## backup-router: Address of router to use while booting
    ##
    backup-router 192.168.100.1;
    root-authentication {
      ##
      ## encrypted-password: Encrypted password string
      ##
      encrypted-password "$1$BYJQE$/ocQof8pmcm7MSGK0"; # SECRET-DATA
    }
    ##
    ## name-server: DNS name servers
    ## require: system
    ##
    name-server {
      ##
      ## name-server: DNS name server address
      ##
      208.197.1.0;
    }
    login {
      ##
      ## class: User name (login)
      ## match: ^[[:alnum:]_-]+$
      ##
```

```
        class super-user {
            ##
            ## permissions: Set of permitted operation categories
            ##
            permissions all;
        }
        ...
        ##
        ## services: System services
        ## require: system
        ##
        services {
            ## services: Service name
            ##
            ftp;
            ##
            ## services: Service name
            ##
            telnet;
            ##
        }
        syslog {
            ##
            ## file-name: File to record logging data
            ##
            file messages {
                ##
                ## Facility type
                ## Level name
                ##
                any notice;
                ##
                ## Facility type
                ## Level name
                ##
            authorization info;
            }
        }
}
chassis {
        alarm {
            sonet {
                ##
                ## lol: Loss of light
                ## alias: loss-of-light
                ##
                lol red;
                }
            }
        }
}
```

```
interfaces {
    ##
    ## Interface name
    ##
    at-2/1/1 {
        atm-options {
            ##
            ## vpi: Virtual path index
            ## range: 0 .. 255
            ## maximum-vcs: Maximum number of virtual circuits on this VP
            ##
            vpi 0 maximum-vcs 512;
        }
        ##
        ## unit: Logical unit number
        ## range: 0 .. 16384
        ##
        unit 0 {
            ##
            ## vci: ATM point-to-point virtual circuit identifier ([vpi.]vci)
            ## match: ^([[:digit:]]+.){0,1}[[:digit:]]+$
            ##
            vci 0.128;
        }
    }
...
```

Chapter 7

Managing Configurations

This chapter provides basic information about managing configurations.

Topics include:

How the Configuration Is Stored

When you edit a configuration, you work in a copy of the current configuration to create a candidate configuration. The changes you make to the candidate configuration are visible in the CLI immediately, so if multiple users are editing the configuration at the same time, all users can see all changes.

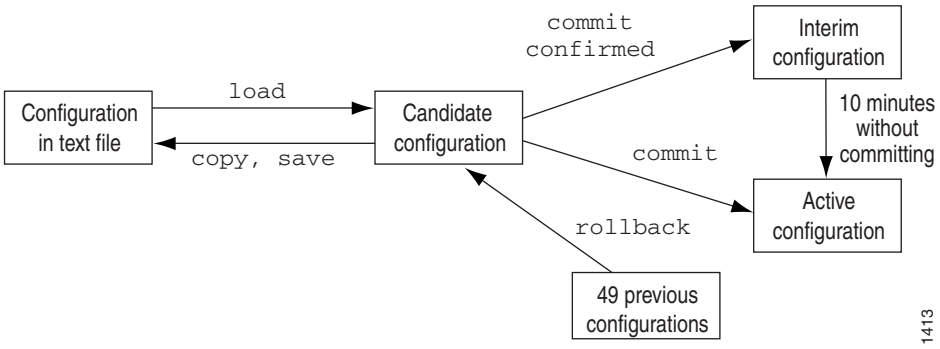
To have a candidate configuration take effect, you *commit* the changes. At this point, the candidate file is checked for proper syntax, activated, and marked as the current, operational software configuration file. If multiple users are editing the configuration, when you commit the candidate configuration, all changes made by all the users take effect.

In addition to saving the current configuration, the CLI saves the current operational version and the previous 49 versions of committed configurations. The most recently committed configuration is version 0 (the current operational version, which is the default configuration that the system returns to if you roll back to a previous configuration), and the oldest saved configuration is version 49.

The currently operational JUNOS software configuration is stored in the file juniper.conf, and the last three committed configurations are stored in the files juniper.conf.1, juniper.conf.2, and juniper.conf.3. These four files are located in the directory /config, which is on the router's flash drive. The remaining 46 previous versions of committed configurations, the files juniper.conf.4 through juniper.conf.49, are stored in the directory /var/db/config on the hard disk.

Figure 16 illustrates the various router configuration states and the configuration mode commands you use to load, commit, copy, save, or roll back the configuration.

Figure 16: Commands for Storing and Modifying the Router Configuration



Returning to the Most Recently Committed Configuration

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

```
[edit]
user@host# rollback

load complete
```

To activate the configuration to which you rolled back, use the commit command:

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

Returning to a Configuration Prior to the Most Recently Committed One

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

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

load complete
```

Displaying Previous Configurations

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

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

Possible completions:
<[Enter]> Execute this command
<number> Numeric argument
  0                    2005-02-27 12:52:10 PST by abc via cli
  1                    2005-02-26 14:47:42 PST by def via cli
  2                    2005-02-14 21:55:45 PST by ghi via cli
  3                    2005-02-10 16:11:30 PST by jkl via cli
  4                    2005-02-10 16:02:35 PST by mno via cli
  5                    2005-03-16 15:10:41 PST by pqr via cli
  6                    2005-03-16 14:54:21 PST by stu via cli
  7                    2005-03-16 14:51:38 PST by vwx via cli
  8                    2005-03-16 14:43:29 PST by yzz via cli
  9                    2005-03-16 14:15:37 PST by abc via cli
  10                   2005-03-16 14:13:57 PST by def via cli
  11                   2005-03-16 12:57:19 PST by root via other
  12                   2005-03-16 10:45:23 PST by root via other
```

```
  13                  2005-03-16 10:08:13 PST by root via other
  14                  2005-03-16 01:20:56 PST by root via other
  15                  2005-03-16 00:40:37 PST by ghi via cli
  16                  2005-03-16 00:39:29 PST by jkl via cli
  17                  2005-03-16 00:32:36 PST by mno via cli
  18                  2005-03-16 00:31:17 PST by pqr via cli
  19                  2005-03-15 19:59:00 PST by stu via cli
  20                  2005-03-15 19:53:39 PST by vwx via cli
  21                  2005-03-15 18:07:19 PST by yzz via cli
  22                  2005-03-15 17:59:03 PST by abc via cli
  23                  2005-03-15 15:05:14 PST by def via cli
  24                  2005-03-15 15:04:51 PST by ghi via cli
  25                  2005-03-15 15:03:42 PST by jkl via cli
  26                  2005-03-15 15:01:52 PST by mno via cli
  27                  2005-03-15 14:58:34 PST by pqr via cli
  28                  2005-03-15 13:09:37 PST by root via other
  29                  2005-03-12 11:01:20 PST by stu via cli
  30                  2005-03-12 10:57:35 PST by vwx via cli
  31                  2005-03-11 10:25:07 PST by yzz via cli
  32                  2005-03-10 23:40:58 PST by abc via cli
  33                  2005-03-10 23:40:38 PST by def via cli
  34                  2005-03-10 23:14:27 PST by ghi via cli
  35                  2005-03-10 23:10:16 PST by jkl via cli
  36                  2005-03-10 23:01:51 PST by mno via cli
  37                  2005-03-10 22:49:57 PST by pqr via cli
  38                  2005-03-10 22:24:07 PST by stu via cli
  39                  2005-03-10 22:20:14 PST by vwx via cli
  40                  2005-03-10 22:16:56 PST by yzz via cli
  41                  2005-03-10 22:16:41 PST by abc via cli
  42                  2005-03-10 20:44:00 PST by def via cli
  43                  2005-03-10 20:43:29 PST by ghi via cli
  44                  2005-03-10 20:39:14 PST by jkl via cli
  45                  2005-03-10 20:31:30 PST by root via other
  46                  2005-03-10 18:57:01 PST by mno via cli
  47                  2005-03-10 18:56:18 PST by pqr via cli
  48                  2005-03-10 18:47:49 PST by stu via cli
  49                  2005-03-10 18:47:34 PST by vw via cli
|   Pipe through a command
[edit]
```

Comparing Configuration Changes with a Prior Version

In configuration mode only, when you have made changes to the configuration and want to compare the candidate configuration with a prior version, you can use the compare command to display the configuration. The compare command compares the candidate configuration with either the current committed configuration or a configuration file and displays the differences between the two configurations. To compare configurations, specify the compare command after the pipe:

```
[edit]
user@host# show | compare [filename | rollback n]
```

filename is the full path to a configuration file. The file must be in the proper format: a hierarchy of statements.

n is the index into the list of previously committed configurations. The most recently saved configuration is number 0, and the oldest saved configuration is number 49. If you do not specify arguments, the candidate configuration is compared against the active configuration file (/config/juniper.conf).

The comparison output uses the following conventions:

- Statements that are only in the candidate configuration are prefixed with a plus sign (+).
- Statements that are only in the comparison file are prefixed with a minus sign (–).
- Statements that are unchanged are prefixed with a single blank space ( ).

The following example shows various changes, then a comparison of the candidate configuration with the active configuration, showing only the changes made at the [edit protocols bgp] hierarchy level:

```
[edit]
user@host# edit protocols bgp

[edit protocols bgp]
user@host# show
group my-group {
    type internal;
    hold-time 60;
    advertise-inactive;
    allow 1.1.1.1/32;
}
group fred {
    type external;
    peer-as 33333;
    allow 2.2.2.2/32;
}
group test-peers {
    type external;
    allow 3.3.3.3/32;
}
[edit protocols bgp]
user@host# set group my-group hold-time 90
```

```
[edit protocols bgp]
user@host# delete group my-group advertise-inactive
[edit protocols bgp]
user@host# set group fred advertise-inactive
[edit protocols bgp]
user@host# delete group test-peers
[edit protocols bgp]
user@host# show | compare
[edit protocols bgp group my-group]
-     hold-time 60;
+     hold-time 90;
-     advertise-inactive;
[edit protocols bgp group fred]
+     advertise-inactive;
[edit protocols bgp]
-group test-peers {
-     type external;
-     allow 3.3.3.3/32;
-}
[edit protocols bgp]
user@host# show
group my-group {
     type internal;
     hold-time 90;
     allow 1.1.1.1/32;
}
group fred {
     type external;
     advertise-inactive;
     peer-as 3333;
     allow 2.2.2.2/32;
}
```

Creating and Returning to a Rescue Configuration

A *rescue* configuration allows you to define a known working configuration or a configuration with a known state that you can roll back to at any time. This alleviates the necessity of having to remember the rollback number with the rollback command. You use the rescue configuration when you need to roll back to a known configuration or as a last resort if your router configuration and the backup configuration files become damaged beyond repair.

To save the most recently committed configuration as the rescue configuration so that you can return to it at any time, issue the request system configuration rescue save command:

```
user@host> request system configuration rescue save
user@host>
```

To return to the rescue configuration, use the rollback rescue configuration mode command:

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

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

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

To save the most recently committed configuration as the rescue configuration so that you can return to it at any time using the rollback command, issue the request system configuration rescue save command:

```
user@host> request system configuration rescue save
user@host>
```

To delete an existing rescue configuration, issue the request system configuration rescue delete command:

```
user@host> request system configuration rescue delete
user@host>
```

For more information about the request system configuration rescue delete and request system configuration rescue save commands, see the *JUNOS System Basics and Services Command Reference*.

Saving a Configuration to a File

You might want to save the configuration to a file so that you can edit it with a text editor of your choice. You can save your current configuration to an ASCII file, which saves the configuration in its current form, including any uncommitted changes. If more than one user is modifying the configuration, all changes made by all users are saved.

To save software configuration changes to an ASCII file, use the save configuration mode command:

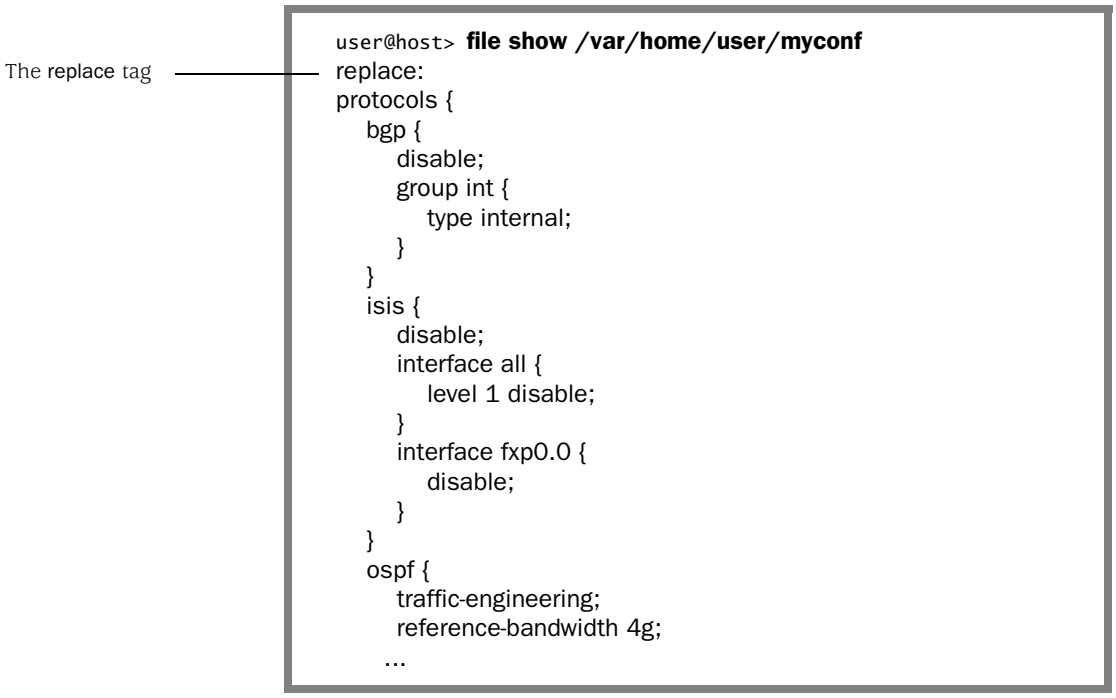
```
[edit]
user@host# save filename
[edit]
user@host#
```

The contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it. This allows a section of the configuration to be saved, while fully specifying the statement hierarchy.

By default, the configuration is saved to a file in your home directory, which is on the flash drive. For information about specifying the filename, see "Specifying Filenames and URLs" on page 60.

When you issue this command from anywhere in the hierarchy (except the top level), a replace tag is automatically included at the beginning of the file. You can use the replace tag to control how a configuration is loaded from a file. (See Figure 17.) For more information, see "Loading a Configuration From a File" on page 117.

Figure 17: The replace Tag



Loading a Configuration From a File

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

You can also create a configuration while typing at the terminal and then load it. Loading a configuration from the terminal is generally useful when you are cutting existing portions of the configuration and pasting them elsewhere in the configuration.

To load an existing configuration file that is located on the router, use the load configuration mode command:

```
[edit]
user@host# load (merge | override | patch | replace | set | update) filename
    <relative>
```

To load a configuration from the terminal, use the following version of the load configuration mode command:

```
[edit]
user@host# load (merge | override | patch | replace | set | update) terminal
    <relative>
[Type ^D to end input]
```

To replace an entire configuration, specify the override option at any level of the hierarchy.

An override operation discards the current candidate configuration and loads the configuration in *filename* or the one that you type at the terminal. When you use the override option and commit the configuration, all system processes reparse the configuration. For an example, see Figure 18 on page 119.

To replace only the configuration that has changed, specify the update option at any level of the hierarchy. An update operation compares the current configuration and the current candidate configuration, and loads only the changes between these configurations in *filename* or the one that you type at the terminal. When you use the update operation and commit the configuration, the JUNOS software attempts to notify the smallest set of system processes that are affected by the configuration change.

To combine the current configuration and the configuration in *filename* or the one that you type at the terminal, specify the merge option. A merge operation is useful when you are adding a new section to an existing configuration. If the existing configuration and the incoming configuration contain conflicting statements, the statements in the incoming configuration override those in the existing configuration. For an example, see Figure 20 on page 120.

To replace portions of a configuration, specify the **replace** option. For this operation to work, you must include **replace:** tags in the file or configuration you type at the terminal. The software searches for the **replace:** tags, deletes the existing statements of the same name, if any, and replaces them with the incoming configuration. If there is no existing statement of the same name, the **replace** operation adds to the configuration the statements marked with the **replace:** tag. For an example, see Figure 19 on page 119.

To load a configuration that contains the **set** configuration mode command, specify the **set** option. This option executes the configuration instructions line by line as they are stored in a file or from a terminal. The instructions can contain any configuration mode command, such as **set**, **edit**, **exit**, and **top**. For an example, see Figure 22 on page 121.

To use the **merge**, **replace**, **set**, or **update** option without specifying the full hierarchy level, specify the **relative** option. For example:

```
[edit system]
user@host# show static-host-mapping
bob sysid 987.654.321ab

[edit system]
user@host# load replace terminal relative
{Type ^D at a new line to end input]
replace: static-host-mapping {
    bob sysid 0123.456.789bc;
}
load complete

[edit system]
user@host# show static-host-mapping
bob sysid 0123.456.789bc;
```

To change part of the configuration with a patch file and mark only those parts as changed, specify the **patch** option. For an example, see Figure 21 on page 120.

If, in an override or merge operation, you specify a file or type text that contains **replace:** tags, the **replace:** tags are ignored, and the override or merge operation is performed.

If you are performing a **replace** operation and the file you specify or text you type does not contain any **replace:** tags, the replace operation is effectively equivalent to a **merge** operation. This might be useful if you are running automated scripts and cannot know in advance whether the scripts need to perform a replace or a **merge** operation. The scripts can use the **replace** operation to cover either case.

For information about specifying the filename, see "Specifying Filenames and URLs" on page 60.

To copy a configuration file from another network system to the local router, you can use the SSH and telnet utilities, as described in the *JUNOS System Basics and Services Command Reference*.

Examples: Loading a Configuration from a File

Figure 18: Example 1: Load a Configuration from a File

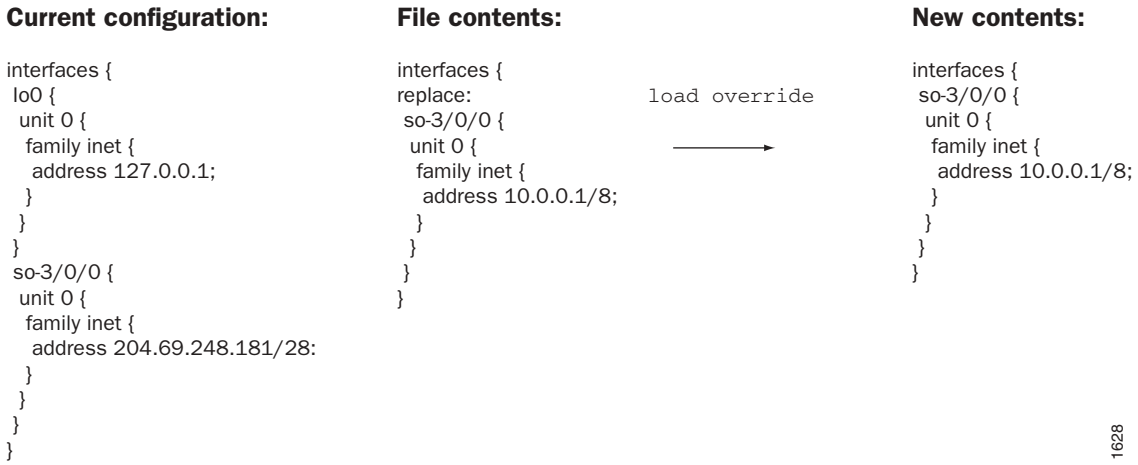


Figure 19: Example 2: Load a Configuration from a File

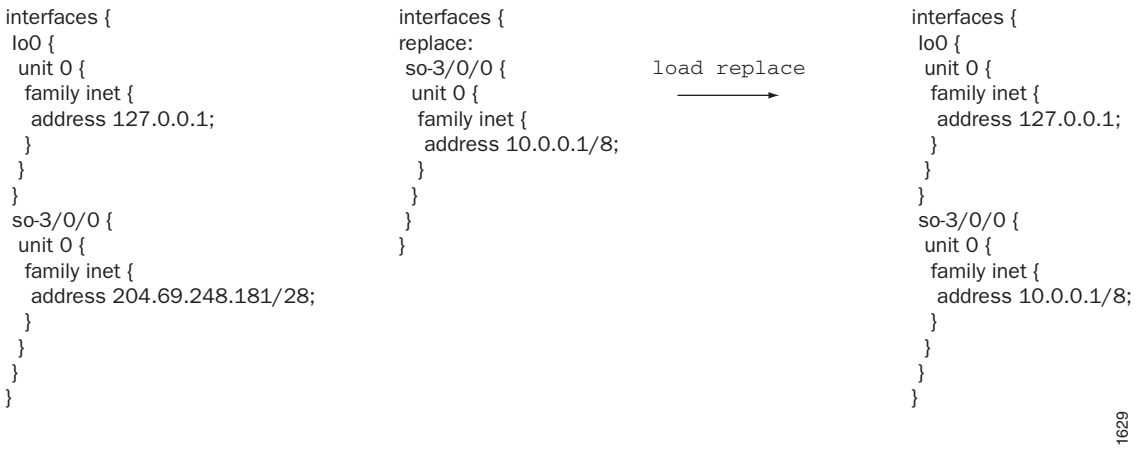


Figure 20: Example 3: Load a Configuration from a File

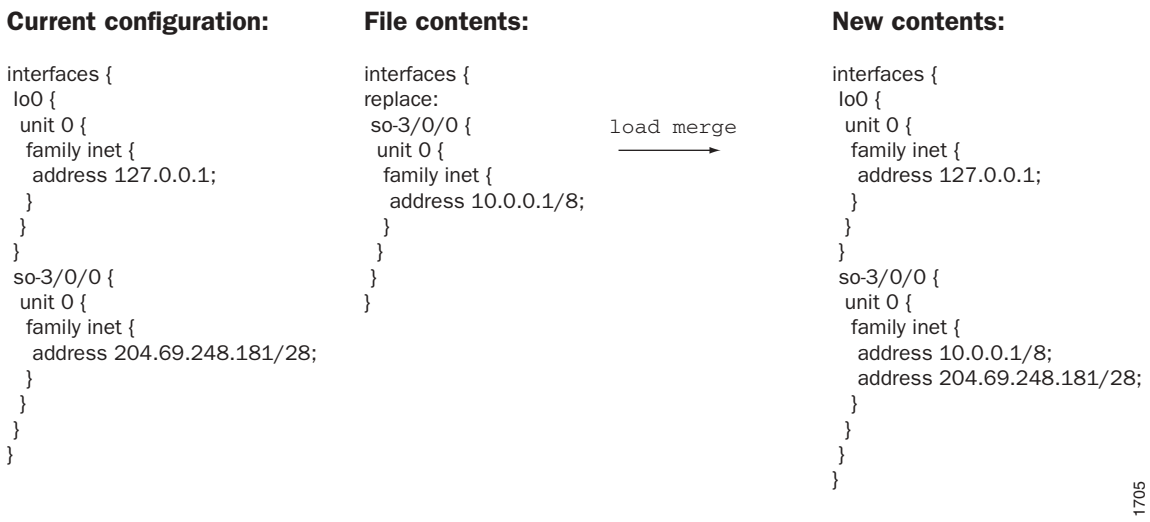


Figure 21: Example 4: Load a Configuration from a File

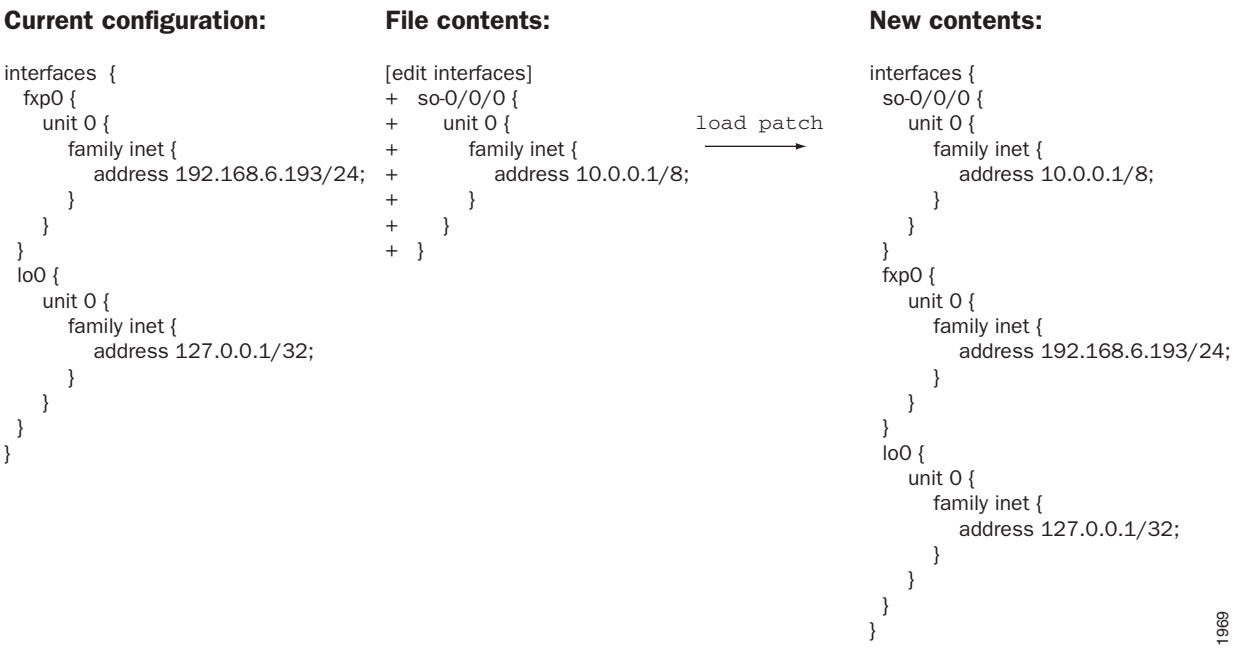


Figure 22: Example 5: Load a Configuration from a File

File contents:

```
edit access
set profile p1 client cl ike
edit profile p1 client cl ike
set pre-shared-key ascii-text "abcd"
set allowed-proxy-pair local 1.1.1.1 remote 2.2.2.2
exit
deactivate profile p1
top
edit system
set radius-server 1.1.1.1
```

```
load set
```


New contents:

```
system {
  radius-server {
    1.1.1.1;
  }
}
access {
  inactive: profile p1 {
    client cl {
      ike {
        allowed-proxy-pair local 1.1.1.1/32 remote 2.2.2.2/32;
        pre-shared-key ascii-text "$9$Ydg4ZDjqf5FVw"; ## SECRET-DATA
      }
    }
  }
}
```

Additional Details About Specifying Statements and Identifiers

This section provides more detailed information about CLI container and leaf statements so that you can better understand how you must specify them when creating ASCII configuration files. It also describes how the CLI performs type-checking to verify that the data you entered is in the correct format.

Specifying Statements

Statements are shown one of two ways, either with braces or without:

- Statement name and identifier, with one or more lower-level statements enclosed in braces:

  < *statement-name* > < *identifier* > {
  statement;
  additional-statements;
  }

- Statement name, identifier, and a single identifier:

  < *statement-name* > < *identifier* > *identifier*;

The *statement-name* is the name of the statement. In the configuration example shown in the previous section, ospf and area are statement names.

The *identifier* is a name or other string that uniquely identifies an instance of a statement. The identifier is used when a statement can be specified more than once in a configuration. In the configuration example shown in the previous section, the identifier for the area statement is 0 and the identifier for the interface statement is so-0/0/0.

When specifying a statement, you must specify either a statement name or an identifier, or both, depending on the statement hierarchy.

You specify identifiers in one of the following ways:

- *identifier*—The *identifier* is a flag, which is a single keyword.
- *identifier value*—The *identifier* is a keyword, and the *value* is a required option variable.
- *identifier* [*value1 value 2 value3 ...*]—The *identifier* is a set that accepts multiple values. The brackets are required when you specify a set of identifiers; however, they are optional when you specify only one identifier.

The following examples illustrate how statements and identifiers are specified in the configuration:

```
protocol {                  # Top-level statement (statement-name).
    ospf {                  # Statement under "protocol" (statement-name).
        area 0.0.0.0 {      # OSPF area "0.0.0.0" (statement-name identifier),
            interface so-0/0/0 {# which contains an interface named "so-0/0/0."
                hello-interval 25;# Identifier and value (identifier-name value).
                priority 2;     # Identifier and value (identifier-name value).
                disable;        # Flag identifier (identifier-name).
            }
            interface so-0/0/1;# Another instance of "interface," named so-0/0/1,
        }                       # this instance contains no data, so no braces
    }                           # are displayed.
}
policy-options {            # Top-level statement (statement-name).
    term term1 {            # Statement under "policy-options"
                            # (statement-name value).
        from {              # Statement under "term" (statement-name).
            route-filter 10.0.0.0/8 orlonger reject;# One identifier ("route-filter") with
            route-filter 127.0.0.0/8 orlonger reject;# multiple values.
            route-filter 128.0.0.0/16 orlonger reject;
            route-filter 149.20.64.0/24 orlonger reject;
            route-filter 172.16.0.0/12 orlonger reject;
            route-filter 191.255.0.0/16 orlonger reject;
        }
        then {              # Statement under "term" (statement-name).
            next term;      # Identifier (identifier-name).
        }
    }
}
```

When you create an ASCII configuration file, you can specify statements and identifiers in one of the following ways. However, each statement has a preferred style, and the CLI uses that style when displaying the configuration in response to a configuration mode show command.

- Statement followed by identifiers:

  statement-name identifier-name [...] *identifier-name value* [...];

- Statement followed by identifiers enclosed in braces:

  statement-name {
      identifier-name;
      [...]
      identifier-name value;
      [...]
  }

- For some repeating identifiers, you can use one set of braces for all the statements:

  statement-name {
      identifier-name value1;
      identifier-name value2;
  }

Performing CLI Type-Checking

When you specify identifiers and values, the CLI expects to receive specific types of input and performs type-checking to verify that the data you entered is in the correct format. For example, for a statement in which you must specify an IP address, the CLI checks that you entered an address in a valid format. If you have not, an error message indicates what you were expected to type. Table 13 lists the data types the CLI checks.

Table 13: CLI Configuration Input Types (1 of 2)

Data Type	Format	Examples
Physical interface name (used in the [edit interfaces] hierarchy)	*type-fpc/pic/port*	**Correct:** so-0/0/1 **Incorrect:** so-0
Full interface name	*type-fpc/pic/port<:channel>.logical*	**Correct:** so-0/0/1.0 **Incorrect:** so-0/0/1
Full or abbreviated interface name (used in places other than the [edit interfaces] hierarchy)	*type-<fpc</pic/port>><<:channel>.logical>*	**Correct:** so, so-1, so-1/2/3:4.5
IP address	0x*hex-bytes* *octet<.octet<.octet.<octet>>>*	**Correct:** 1.2.3.4, 0x01020304, 128.8.1, 128.8 **Sample translations:** 1.2.3 becomes 1.2.3.0 0x01020304 becomes 1.2.3.4 0x010203 becomes 0.1.2.3

Table 13: CLI Configuration Input Types (2 of 2)

Data Type	Format	Examples
IP address (destination prefix) and prefix length	0x*hex-bytes</length>* *octet<.octet<.octet.<octet>>></length* >	**Correct:** 10/8, 128.8/16, 1.2.3.4/32, 1.2.3.4 **Sample translations:** 1.2.3 becomes 1.2.3.0/32 0x01020304 becomes 1.2.3.4/32 0x010203 becomes 0.1.2.3/32 default becomes 0.0.0.0/0
International Organization for Standardization (ISO) address	*hex-nibble<hex-nibble ...>*	**Correct:** 47.1234.2345.3456.00, 47123423453456.00, 47.12.34.23.45.34.56.00 **Sample translations:** 47123456 becomes 47.1234.56 47.12.34.56 becomes 47.1234.56 4712.3456 becomes 47.1234.56
OSPF area identifier (ID)	0x*hex-bytes* *octet<.octet<.octet.<octet>>>* d*ecimal-number*	**Correct:** 54, 0.0.0.54, 0x01020304, 1.2.3.4 **Sample translations:** 54 becomes 0.0.0.54 257 becomes 0.0.1.1 128.8 becomes 128.8.0.0 0x010203 becomes 0.1.2.3

Synchronizing Routing Engines

If your router has two Routing Engines, you can manually direct one Routing Engine to synchronize its configuration with the other by issuing the commit synchronize command. The Routing Engine on which you execute this command (requesting Routing Engine) copies and loads its candidate configuration to the other (responding Routing Engine). Both Routing Engines then perform a syntax check on the candidate configuration file being committed. If no errors are found, the configuration is activated and becomes the current operational configuration on both Routing Engines. The commit synchronize command does not work if the responding Routing Engine has uncommitted configuration changes.

For example, if you are logged in to re1 (requesting Routing Engine) and you want re0 (responding Routing Engine) to have the same configuration as re1, issue the commit synchronize command on re1. re1 copies and loads its candidate configuration to re0. Both Routing Engines then perform a syntax check on the candidate configuration file being committed. If no errors are found, re1's candidate configuration is activated and becomes the current operational configuration on both Routing Engines.

NOTE: When you issue the commit synchronize command, you must use the groups re0 and re1. For information about how to use the apply groups statement, see "Applying a Configuration Group" on page 157.

The responding Routing Engine must be running JUNOS Release 5.0 or higher.

For information about issuing the commit synchronize command on a routing matrix, see the *JUNOS System Basics Configuration Guide*.

To synchronize a Routing Engine's current operational configuration file with the other, log in to the Routing Engine from which you want to synchronize and issue the commit synchronize command:

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

NOTE: You can also add the commit synchronize statement at the [edit system] hierarchy level so that a commit command automatically invokes a commit synchronize command by default. For more information, see the *JUNOS System Basics Configuration Guide*.

Example: Using Apply Groups re0 and re1

The following example shows apply groups re0 and re1 with some configuration data that might be different on re0 and re1:

```
re0 {
    system {
        host-name my_router_RE0;
    }
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 192.168.15.49/24;
                }
                family iso;
            }
        }
    }
}
re1 {
    system {
        host-name my_router_RE1;
    }
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 192.168.15.50/24;
                }
                family iso;
            }
        }
    }
}
```

Example: Setting Apply Groups re0 and re1

The following example sets the apply groups re0 and re1:

```
[edit]
user@host# set apply-groups [re0 re1]
[edit]
user@host#
```

Chapter 8
Filtering Command Output

For commands that display output, such as the show commands, you can filter the output. This chapter provides information about the following topics:

Using Keyboard Sequences at the —(more)— Prompt

If the output from a command is longer than the screen length, it appears one screen at a time by means of a UNIX more-type interface. The prompt —(more)— indicates that more output is available. This format is helpful when you want to scroll and search through lengthy output.

As soon as the command-line interface (CLI) can determine the length of the output (typically by the second screen), it displays the percentage of the output that has been displayed so far above the —(more)— prompt.

The output buffer for the prompt is restricted to 32 megabytes (MB). Any new data that exceeds the buffer limit replaces the oldest data in the memory buffer. When the buffer limit is exceeded, attempts to search backward or navigate to the beginning of the output generate a warning indicating that the output is truncated.

Because of the buffer size restriction, use of the scroll and search functions might be limited.

Table 14 lists the keyboard sequences you can use at the –(more)– prompt.

Table 14: –(more)– Prompt Keyboard Sequences

Category	Action	Keyboard Sequence
Get Help	Display information about the keyboard sequences you can display at the –(more)– prompt.	h
Scroll Down	Scroll down one line.	Enter, k, Ctrl+m, Ctrl+n, or down arrow
	Scroll down one-half screen.	Tab, d, Ctrl+d, or Ctrl+x
	Scroll down one whole screen.	Space or Ctrl+f
	Scroll down to the bottom of the output.	Ctrl+e or G
	Display the output all at once instead of one screen at a time. (Same as specifying the | no-more filter when entering commands using the pipe symbol. See "Preventing Output from Being Paginated" on page 134.)	N
Scroll Up	Display the previous line of output.	j, Ctrl+h, Ctrl+p, or up arrow
	Scroll up one-half screen.	u or Ctrl+u
	Scroll up one whole screen.	b or Ctrl+b
	Scroll up to the top of the output.	Ctrl+a or g
Search	Search forward for a string.	/*string*
	Search backward for a string.	?*string*
	Repeat the previous search for a string.	n
	Search for a text string. You are prompted for the string to match. (Same as specifying the | match filter when entering command using the pipe symbol. See "Displaying Output That Matches a Regular Expression" on page 134.)	m or M
	Search, ignoring a text string. You are prompted for the string to not match. (Same as specifying the | except filter when entering command using the pipe symbol. See "Ignoring Output That Does Not Match a Regular Expression" on page 133.)	e or E
Interrupt or End Output, Redraw the Output, and Save the Output to a File	Interrupt the display of output.	Ctrl+c, q, Q, or Ctrl+k
	Do not redisplay the CLI prompt immediately after displaying the output, but remain at the –(more)– prompt. (Same as specifying the | hold command.)	H
	Clear any match conditions and display the complete output.	c or C
	Redraw the output on the screen.	Ctrl+l
	Save the command output to a file. You are prompted for a filename. (Same as specifying the | save *filename* command.)	s or S

Using the Pipe (|) Symbol When Entering Commands

You can filter output by adding the | (or *pipe*) symbol when you enter a command. For example:

```
user@host> show rip neighbor ?
Possible completions:
  <[Enter]>            Execute this command
  <name>               Name of RIP neighbor
  instance             Name of RIP instance
  logical-router       Name of logical router, or 'all'
  |                    Pipe through a command
```

The following example lists the filters that can be used with the pipe symbol:

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

For the show configuration command only, an additional compare filter is available:

```
user@host> show configuration | ?
Possible completions:
  compare              Compare configuration changes with prior version
...
```

You can enter any of the pipe filters in conjunction. For example:

```
user@host> command | match regular-expression | save filename
```

See “Pipe Filter Functions” on page 131 for a description of each type of filter.

NOTE: This section describes *only* the filters that can be used for operational mode command output. For information about filters that can be used in configuration mode, see the *JUNOS System Basics Configuration Guide*.

Using Regular Expressions with the Pipe Symbol

The except, find, and match filters used with the pipe symbol employ regular expressions to filter output. Juniper Networks uses the regular expressions as defined in POSIX 1003.2. (See Table 15.) If the regular expression contains spaces, operators, or wildcard characters, enclose the expression in quotation marks.

Table 15: Common Regular Expression Operators in Operational Mode Commands

Operator	Function
|	Indicates that a match can be one of the two terms on either side of the pipe.
^	Used at the beginning of an expression, denotes where a match should begin.
$	Used at the end of an expression, denotes that a term must be matched exactly up to the point of the $ character.
[ ]	Specifies a range of letters or digits to match. To separate the start and end of a range, use a hyphen (-).
( )	Specifies a group of terms to match.

For example, if a command produces the following output:

```
1 2
2 2
3 2 1
4
```

a pipe filter of | match 2 displays the following output:

```
1 2
2 2
3 2 1
```

and a pipe filter of | except 1 displays the following output:

```
2 2
4
```

NOTE: See the following sections for more examples of using regular expressions:

- "Ignoring Output That Does Not Match a Regular Expression" on page 133
- "Displaying Output from the First Match of a Regular Expression" on page 133
- "Displaying Output That Matches a Regular Expression" on page 134

Pipe Filter Functions

This section describes each pipe filter:

Comparing Configurations

The compare filter compares the candidate configuration with either the current committed configuration or a configuration file and displays the differences between the two configurations. To compare configurations, enter compare after the pipe symbol:

```
[edit]
user@host# show | compare [filename | rollback n]
```

filename is the full path to a configuration file.

n is the index into the list of previously committed configurations. The most recently saved configuration is 0. If you do not specify arguments, the candidate configuration is compared against the active configuration file (/config/juniper.conf).

The comparison output uses the following conventions:

- Statements that are only in the candidate configuration are prefixed with a plus sign (+).
- Statements that are only in the comparison file are prefixed with a minus sign (–).
- Statements that are unchanged are prefixed with a single blank space ( ).

For example:

```
user@host> show configuration system | compare rollback 9
[edit system]
+ host-name nutmeg;
+ backup-router 192.168.71.254;
- ports {
-     console log-out-on-disconnect;
- }
[edit system name-server]
+  172.17.28.11;
   172.17.28.101 { ... }
[edit system name-server]
   172.17.28.101 { ... }
+  172.17.28.100;
+  172.17.28.10;
[edit system]
-  scripts {
-      commit {
-          allow-transients;
-      }
-  }
+  services {
+      ftp;
+      rlogin;
+      rsh;
+      telnet;
+  }
```

Counting the Number of Lines of Output

To count the number of lines in the output from a command, enter count after the pipe symbol. For example:

```
user@host> show configuration | count
Count: 269 lines
```

Displaying Output in XML Tag Format

To display command output in XML tag format, enter display xml after the pipe symbol.

The following example displays the show cli directory command output as XML tags:

```
user@host> show cli directory | display xml
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/7.5I0/junos">
    <cli>
        <working-directory>/var/home/regress</working-directory>
    </cli>
    <cli>
        <banner></banner>
    </cli>
</rpc-reply>
```

Ignoring Output That Does Not Match a Regular Expression

To ignore text that matches a regular expression, specify the except command after the pipe symbol. If the regular expression contains any spaces, operators, or wildcard characters, enclose it in quotation marks. For information on common regular expression operators, see Table 15 on page 130.

The following example displays all users who are logged in to the router, except for the user root:

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

Displaying Output from the First Match of a Regular Expression

To display output starting with the first occurrence of text matching a regular expression, enter find after the pipe symbol. If the regular expression contains any spaces, operators, or wildcard characters, enclose it in quotation marks. For information on common regular expression operators, see Table 15 on page 130.

The following example displays the routes in the routing table starting at IP address 208.197.169.0:

```
user@host> show route | find 208.197.169.0
208.197.169.0/24   *[Static/5] 1d 13:22:11
                    > to 192.168.4.254 via so-3/0/0.0
224.0.0.5/32       *[OSPF/10] 1d 13:22:12, metric 1

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

47.0005.80ff.f800.0000.0108.0001.1921.6800.4015.00/160
                   *[Direct/0] 1d 13:22:12
                    > via lo0.0
```

The following example displays the first CCC entry in the forwarding table:

```
user@host> show route forwarding-table | find ccc
Routing table: ccc
MPLS:
Interface.Label    Type RtRef Nexthop       Type Index NhRef Netif
default            perm     0               rjct     3     1
0                  user     0               recv     5     2
1                  user     0               recv     5     2
32769              user     0               ucst    45     1 fe-0/0/0.534
fe-0/0/0. (CCC)    user     0               indr    44     2
                              10.0.16.2     Push 32768, Push
```

Retaining Output After the Last Screen

To not return immediately to the CLI prompt after viewing the last screen of output, enter hold after the pipe symbol. The following example prevents returning to the CLI prompt after you have viewed the last screen of output from the show log log-file-1 command:

```
user@host> show log log-file-1 | hold
```

This filter is useful when you want to scroll or search through output.

Displaying Output Beginning with the Last Entries

To display text starting from the end of the output, enter last <*lines*> after the pipe symbol.

The following example displays the last entries in log-file-1 file:

```
user@host> show log log-file-1 | last
```

This filter is useful for viewing log files in which the end of the file contains the most recent entries.

Displaying Output That Matches a Regular Expression

To display output that matches a regular expression, enter match *regular-expression* after the pipe symbol. If the regular expression contains any spaces, operators, or wildcard characters, enclose it in quotation marks. For information on common regular expression operators, see Table 15 on page 130.

The following example matches all the Asynchronous Transfer Mode (ATM) interfaces in the configuration:

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

Preventing Output from Being Paginated

By default, if output is longer than the length of the terminal screen, you are provided with a –(-more)— message to display the remaining output. To display the remaining output, press the spacebar.

To prevent the output from being paginated, enter no-more after the pipe symbol.

The following example displays output from the show configuration command all at once:

```
user@host> show configuration | no-more
```

This feature is useful, for example, if you want to copy the entire output and paste it into an e-mail.

Sending Command Output to Other Users

To display command output on the terminal of a specific user logged in to your router, or on the terminals of all users logged in to your router, enter request message (all | user *account@terminal*) after the pipe symbol.

If you are troubleshooting your router and, for example, talking with a customer service representative on the phone, you can use the request message command to send your representative the command output you are currently viewing on your terminal.

The following example sends the output from the show interfaces command you enter on your terminal to the terminal of the user root@ttyp1:

```
user@host> show interfaces | request message user root@ttyp1
```

The user root@ttyp1 sees the following output appear on the terminal screen:

```
Message from user@host on /dev/ttyp0 at 10:32 PST...
Physical interface: dsc, Enabled, Physical link is Up
   Interface index: 5, SNMP ifIndex: 5
   Type: Software-Pseudo, MTU: Unlimited ...
```

Resolving IP Addresses

If the output of a command displays an unresolved IP address, you can enter | resolve after the command to display the name associated with the IP address. The resolve filter enables the system to perform a reverse DNS lookup of the IP address. If DNS is not enabled, the lookup fails and no substitution is performed.

To perform a reverse DNS lookup of an unresolved IP address, enter resolve <full-names> after the pipe symbol. If you do not specify the full-names option, the name is truncated to fit whatever field width limitations apply to the IP address.

The following example performs a DNS lookup on any unresolved IP addresses in the output from the show ospf neighbors command:

```
user@host> show ospf neighbors | resolve
```

Saving Output to a File

When command output is lengthy, when you need to store or analyze the output, or when you need to send the output in an e-mail or by FTP, you can save the output to a file. By default, the file is placed in your home directory on the router.

To save command output to a file, enter save *filename* after the pipe symbol.

The following example saves the output from the request support information command to a file named my-support-info.txt:

```
user@host> request support information | save my-support-info.txt
Wrote 1143 lines of output to 'my-support-info.txt'
user@host>
```

Trimming Output by Specifying the Starting Column

Output appears on the terminal screen in terms of rows and columns. The first alphanumeric character starting at the left of the screen is in column 1, the second character is in column 2, and so on. To display output starting from a specific column (thus trimming the leftmost portion of the output), enter trim columns after the pipe symbol. The trim filter is useful for trimming the date and time from the beginning of system log messages

The following example displays output from the show system storage command, filtering out the first 10 columns:

```
user@host> show system storage | trim 11
```

Chapter 9

Controlling the CLI Environment

In operational mode, you can control the command-line interface (CLI) environment. For example, you can specify the number lines that are displayed on the screen or your terminal type. The following output lists the options that you can use to control the CLI environment:

```
user@host> set cli ?
Possible completions:
  complete-on-space Toggle word completion on space
  directory         Set the current working directory
  idle-timeout      Set the cli maximum idle time
  prompt            Set the cli command prompt string
  restart-on-upgrade Set cli to prompt for restart after a software upgrade
  screen-length     Set number of lines on screen
  screen-width      Set number of characters on a line
  terminal          Set terminal type
  timestamp         Timestamp cli output
```

NOTE: When you use SSH to log in to the router or log in from the console when its terminal type is already configured (as described in the *JUNOS System Basics Configuration Guide*), your terminal type, screen length, and screen width are already set.

This chapter discusses the following topics:

- Displaying CLI Settings on page 140
- Example: Controlling the CLI Environment on page 140

Setting the Terminal Type

To set the terminal type, use the set cli terminal command:

user@host> **set cli terminal** *terminal-type*

The terminal type can be one of the following: ansi, vt100, small-xterm, or xterm.

Setting the Screen Length

The default CLI screen length is 24 lines. To change the length, use the set cli screen-length command:

user@host> **set cli screen-length** *length*

Setting the screen length to 0 lines disables the display of output one screen at a time. Disabling this UNIX more-type interface can be useful when you are issuing CLI commands from scripts.

Setting the Screen Width

The default CLI screen width is 80 columns. To change the width, use the set cli screen-width command:

user@host> **set cli screen-width** *width*

Setting the CLI Prompt

The default CLI prompt is user@host>. To change this prompt, use the set cli prompt command. If the prompt string contains spaces, enclose the string in quotation marks (" ").

user@host> **set cli prompt** *string*

Setting the CLI Directory

To the set the current working directory, use the set cli directory command:

user@host> **set cli directory** *directory*

directory is the pathname of working directory.

Setting the CLI Timestamp

By default, CLI output does not include a timestamp. To include a timestamp in CLI output, use the set cli timestamp command:

```
user@host> set cli timestamp [format time-date-format | disable]
```

If you do not specify a timestamp format, the default format is *Mmm dd hh:mm:ss* (for example, Feb 08 17:20:49). Enclose the format in single quotation marks (‘).

Setting the Idle Timeout

By default, an individual CLI session never times out after extended times, unless the idle-timeout statement has been included in the user’s login class configuration. To set the maximum time an individual session can be idle before the user is logged off the router, use the set cli idle-timeout command:

```
user@host> set cli idle-timeout timeout
```

timeout can be 0 through 100,000 minutes. Setting *timeout* to 0 disables the timeout.

Setting the CLI to Prompt After a Software Upgrade

By default, the CLI prompts you to restart after a software upgrade. To disable the prompt for an individual session, use the set cli restart-on-upgrade off command:

```
user@host> set cli restart-on-upgrade off
```

To re-enable the prompt, use the set cli restart-on-upgrade on command:

```
user@host> set cli restart-on-upgrade on
```

Setting Command Completion

By default, you can press the spacebar or tab key to have the CLI complete a command.

To have the CLI allow only a tab to complete a command, use the set cli complete-on-space off command:

```
user@host> set cli complete-on-space off
Disabling complete-on-space
user@host>
```

To re-enable the use of both spaces and tabs for command completion, use the set cli complete-on-space on command:

```
user@host> set cli complete-on-space on
Enabling complete-on-space
user@host>
```

Displaying CLI Settings

To display the current CLI settings, use the show cli command:

```
user@host> show cli
CLI screen length set to 24
CLI screen width set to 80
CLI complete-on-space set to on
```

Example: Controlling the CLI Environment

Change the default CLI environment:

```
user@host> set cli screen-length 66
Screen length set to 66
user@host> set cli screen-width 40
Screen width set to 40
user@host> set cli prompt “router1-san-jose > ”
router1-san-jose > show cli
CLI complete-on-space set to on
CLI idle-timeout disabled
CLI restart-on-upgrade set to on
CLI screen length set to 66
CLI screen width set to 40
CLI terminal is ‘xterm’
router1-san-jose >
```

Part 3
Advanced Features

Chapter 10

Using Shortcuts, Wildcards, and Regular Expressions

This chapter provides information on how to use keyboard shortcuts, wildcards, and other advanced techniques to save time when entering commands and configuration statements.

Moving Around and Editing the Command Line

In the CLI, you can use keyboard sequences to move around on a command line and edit the command line. You can also use keyboard sequences to scroll through a list of recently executed commands. Table 16 lists some of the CLI keyboard sequences. They are the same as those used in Emacs.

Table 16: CLI Keyboard Sequences (1 of 2)

Category	Action	Keyboard Sequence
Move the Cursor	Move the cursor back one character.	Ctrl+b
	Move the cursor back one word.	Esc+b or Alt+b
	Move the cursor forward one character.	Ctrl+f
	Move the cursor forward one word.	Esc+f or Alt+f
	Move the cursor to the beginning of the command line.	Ctrl+a
	Move the cursor to the end of the command line.	Ctrl+e
Delete Characters	Delete the character before the cursor.	Ctrl+h, Delete, or Backspace
	Delete the character at the cursor.	Ctrl+d
	Delete all characters from the cursor to the end of the command line.	Ctrl+k
	Delete all characters on the command line.	Ctrl+u or Ctrl+x
	Delete the word before the cursor.	Ctrl+w, Esc+Backspace, or Alt+Backspace
	Delete the word after the cursor.	Esc+d or Alt+d
Insert Recently Deleted Text	Insert the most recently deleted text at the cursor.	Ctrl+y
Redraw the Screen	Redraw the current line.	Ctrl+l
Display Previous Command Lines	Scroll backward through the list of recently executed commands.	Ctrl+p
	Scroll forward through the list of recently executed commands.	Ctrl+n
	Search the CLI history in reverse order for lines matching the search string.	Ctrl+r
	Search the CLI history by typing some text at the prompt, followed by the keyboard sequence. The CLI attempts to expand the text into the most recent word in the history for which the text is a prefix.	Esc+/

Table 16: CLI Keyboard Sequences (2 of 2)

Category	Action	Keyboard Sequence
Display Previous Command Words	Scroll backward through the list of recently entered words in a command line.	Esc+. or Alt+.
Repeat Keyboard Sequences	Specify the number of times to execute a keyboard sequence. *number* can be from 1 through 9.	Esc+*number sequence* or Alt+*number sequence*

Wildcard Characters in Interface Names

You can use wildcard characters in operational commands to specify groups of interface names without having to type each name individually. Table 17 lists the available wildcard characters. You must enclose all wildcard characters except the asterisk (*) in quotation marks (“ ”).

Table 17: Wildcard Characters for Specifying Interface Names

Wildcard Character	Description
* (asterisk)	Match any string of characters in that position in the interface name. For example, so* matches all SONET/SDH interfaces.
“[*character* <*character...*>]”	Match one or more individual characters in that position in the interface name. For example, “so-[03]”* matches all SONET/SDH interfaces in slots 0 and 3.
“[!*character* <*character...*>]”	Match all characters except the ones included in the brackets. For example, “so-[!03]”* matches all SONET/SDH interfaces except those in slots 0 and 3.
“[*character1-character2*]”	Match a range of characters. For example, so-”[0-3]”* matches all SONET/SDH interfaces in slots 0, 1, 2, and 3.
“[!*character1-character2*]”	Match all characters that are not in the specified range of characters. For example, so-”[!0-3]”* matches all SONET/SDH interfaces in slots 4, 5, 6, and 7.

Using Global Replace in a Configuration

To make global changes to variables and identifiers in a configuration, use the replace configuration mode command. This command replaces a pattern in a configuration with another pattern. For example, you can use this command to find and replace all occurrences of an interface name when a PIC is moved to another slot in the router.

user@host> **replace pattern** *pattern1* **with** *pattern2* <**upto** *n*>

pattern *pattern1* is a text string or regular expression that defines the identifiers and values you want to replace in the configuration.

pattern2 is a text string or regular expression that replaces the identifiers and values located with *pattern1*.

Juniper Networks uses standard UNIX-style regular expression syntax (as defined in POSIX 1003.2). If the regular expression contains spaces, operators, or wildcard characters, enclose the expression in quotation marks. Greedy qualifiers (match as much as possible) are supported. Lazy qualifiers (match as little as possible) are not.

upto *n* specifies the number of objects replaced. The value of *n* controls the total number of objects that are replaced in the configuration (not the total number of times the pattern occurs). Objects at the same hierarchy level (siblings) are replaced first. Multiple occurrences of a pattern within a given object are considered a single replacement. For example, if a configuration contains a 010101 text string, the following command:

```
replace pattern 01 with pattern 02 upto 2
```

replaces 010101 with 020202 (instead of 020201). Replacement of 010101 with 020202 is considered a single replacement (*n* = 1), not three separate replacements (*n* = 3).

If you do not specify an upto option, all identifiers and values in the configuration that match *pattern1* are replaced.

The replace command is available in configuration mode at any hierarchy level. All matches are case-sensitive.

Table 18 shows some common regular expressions you can use with the replace command. Table 19 provides some examples of pattern replacement.

Table 18: Common Regular Expressions

Operator	Function
|	Indicates that a match can be one of the two terms on either side of the pipe.
^	Used at the beginning of an expression, denotes where a match should begin.
$	Used at the end of an expression, denotes that a term must be matched exactly up to the point of the $ character.
[ ]	Specifies a range of letters or digits to match. To separate the start and end of a range, use a hyphen (-).
( )	Specifies a group of terms to match. Stored as numbered variables. Use for back references as \1 \2 .... \9.
*	0 or more terms.
+	One or more terms.
.	Any character except for a space " ".
\	A backslash escapes special characters to suppress their special meaning. For example, \. matches **.** (period symbol).
n	Back reference. Matches the *n*th group.
&	Back reference. Matches the entire match.

Table 19: Replacement Examples

Command	Result
replace pattern myrouter with router1	Match: myrouter Result: router1
replace pattern "192\.168\.(.*)/24" with "10.2.\1/28"	Match: 192.168.3.4/24 Result: 10.2.3.4/28
replace pattern "1.\1" with "abc&def"	Match: 1.1 Result: abc1.1def
replace pattern 1.1 with "abc\&def"	Match: 1#1 Result: abc&def

Example 1: Using Global Replace in a Configuration

Replace an interface name in a configuration:

```
[edit]
user@host# show
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0 {
                hello-interval 5;
            }
        }
    }
}
[edit]
user@host# replace so-0/0/0 with so-1/1/0

[edit]
user@host# show
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-1/1/0 {
                hello-interval 5;
            }
        }
    }
}
```

Example 2: Using Global Replace in a Configuration

Use the *n* back reference to replace a pattern:

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0;
}
fe-3/0/1 {
    vlan-tagging;
    unit 0 {
      description "inet6 configuration. IP: 2000::c0a8::1bf5";
      vlan-id 100;
      family inet {
        address 17.10.1.1/24;
      }
      family inet6 {
        address 2000::c0a8:1bf5/3;
      }
    }
}

[edit]
user@host# replace pattern "(.*):1bf5" with "\11bf5"

[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0;
}
fe-3/0/1 {
    vlan-tagging;
    unit 0 {
      description "inet6 configuration. IP: 2000::c0a8:1bf5";
      vlan-id 100;
      family inet {
        address 17.10.1.1/24;
      }
      family inet6 {
        address 2000::c0a8:1bf4/3;
      }
    }
}
```

The pattern 2000::c0a8::1bf5 is replaced with 2000::c0a8:1bf5.

Example 3: Using Global Replace in a Configuration

Consider the hierarchy shown in Figure 23 on page 150. The text string 010101 appears in three places (description sections of ge-0/0/0, ge-0/0/0.0, and fe-0/0/1). These three instances are three objects.

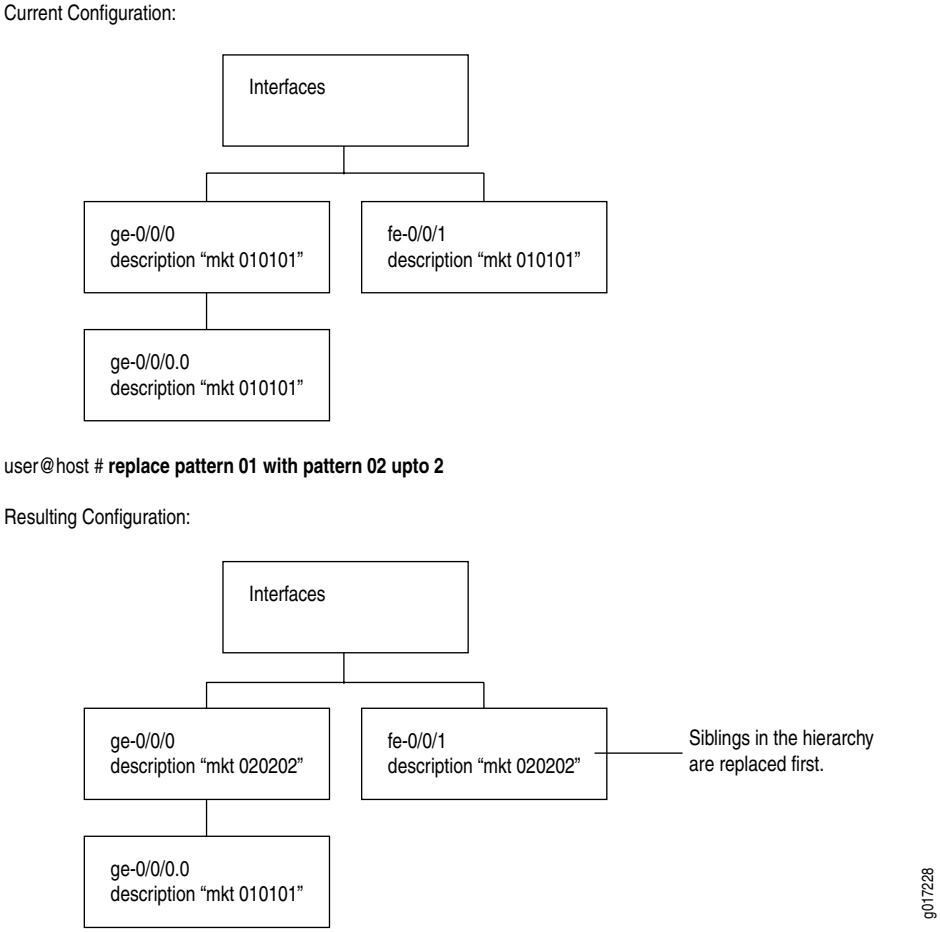
Use the opto option to perform a replacement:

```
user@host# show interfaces
ge-0/0/0 {
description “mkt 010101”; #1st instance in the hierarchy
unit 0 {
description “mkt 010101”; #3rd instance in the hierarchy (child of the 1st
instance)
}
}
fe-0/0/1 {
description “mkt 010101”; #2nd instance in the hierarchy (sibling of the 1st
instance)
unit 0 {
family inet {
address 200.200.20.2/24;
}
}
}
[edit]
user@host# replace pattern 01 with 02 upto 2
[edit]
user@host# commit
commit complete
```

An upto 2 option in the replace command converts 01 to 02 for two object instances. The objects under the main interfaces ge-0/0/0 and fe-0/0/1 will be replaced first (since these are siblings in the hierarchy level). Because of the upto 2 restriction, the replace command replaces patterns in the first and second instance in the hierarchy (siblings), but not the third instance (child of the first instance).

```
[edit]
user@host# show interfaces
ge-0/0/0 {
description “mkt 020202”; #1st instance in the hierarchy
unit 0 {
description “mkt 010101”; #3rd instance in the hierarchy (child of the 1st
instance)
}
}
fe-0/0/1 {
description “mkt 020202”; #2nd instance in the hierarchy (sibling of the 1st
instance)
unit 0 {
family inet {
address 200.200.20.2/24;
}
}
}
```

Figure 23: Replacement by Object



Using Regular Expressions to Remove Related Configuration Items

You can delete related configuration items simultaneously, such as channelized interfaces or static routes, by using a single command and regular expressions. Deleting a statement or an identifier effectively "unconfigures" the functionality associated with that statement or identifier, returning that functionality to its default condition.

You can only delete several parts of the configuration where you normally put multiple items; for example, interfaces. However, you cannot delete "groups" of different items; for example:

```
user@host# show system services
ftp;
rlogin;
rsh;
ssh {
    root-login allow;
}
telnet;

[edit]
user@host# wildcard delete system services *
                                          ^
syntax error.
```

When you delete a statement, the statement and all its subordinate statements and identifiers are removed from the configuration.

To delete related configuration items, issue the wildcard configuration mode command with the delete option and specify the statement path, the items to be summarized with a regular expression, and the regular expression.

```
user@host# wildcard delete <statement-path> <identifier> <regular-expression>
```

NOTE: When you use the wildcard command to remove related configuration items, the regular expression must be the final statement.

If the JUNOS software matches more than eight related items, the CLI displays only the first eight items.

Example: Deleting Interfaces from the Configuration

Delete multiple T1 interfaces in the range from t1-0/0/0:0 through t1-0/0/0:23:

```
user@host# wildcard delete interfaces t1-0/0/0:.*
  matched: t1-0/0/0:0
  matched: t1-0/0/0:1
  matched: t1-0/0/0:2
Delete 3 objects? [yes,no] (no) no
```

Example: Deleting Routes from the Configuration

Delete static routes in the range from 172.0.0.0 to 172.255.0.0:

```
user@host# wildcard delete routing-options static route 172.*
  matched: 172.16.0.0/12
  matched: 172.16.14.0/24
  matched: 172.16.100.0/24
  matched: 172.16.128.0/19
  matched: 172.16.160.0/24
  matched: 172.17.12.0/23
  matched: 172.17.24.0/23
  matched: 172.17.28.0/23
  ...
Delete 13 objects? [yes,no] (no)
```

Chapter 11

Configuration Groups

This chapter discusses the following topics:

Overview

Configuration groups allow you to create a group containing configuration statements and to direct the inheritance of that group's statements in the rest of the configuration. The same group can be applied to different sections of the configuration, and different sections of one group's configuration statements can be inherited in different places in the configuration.

Configuration groups allow you to create smaller, more logically constructed configuration files, making it easier to configure and maintain the JUNOS software. For example, you can group statements that are repeated in many places in the configuration, such as when configuring interfaces, and thereby limit updates to just the group.

You can also use wildcards in a configuration group to allow configuration data to be inherited by any object that matches a wildcard expression.

The configuration group mechanism is separate from the grouping mechanisms used elsewhere in the configuration, such as Border Gateway Protocol (BGP) groups. Configuration groups provide a generic mechanism that can be used throughout the configuration but that are known only to the JUNOS software command-line interface (CLI). The individual software processes that perform the actions directed by the configuration receive the expanded form of the configuration; they have no knowledge of configuration groups.

Inheritance Model

Configuration groups use true inheritance, which involves a dynamic, ongoing relationship between the source of the configuration data and the target of that data. Data values changed in the configuration group are automatically inherited by the target. The target need not contain the inherited information, although the inherited values can be overridden in the target without affecting the source from which they were inherited.

This inheritance model allows you to see only the instance-specific information without seeing the inherited details. A command pipe in configuration mode allows you to display the inherited data.

Configuration Groups Configuration Statements

To configure configuration groups and inheritance, you can include the groups statement at the [edit] hierarchy level:

```
[edit]
groups {
    group-name {
        configuration-data;
    }
}
```

Include the apply-groups [*group-names*] statement anywhere in the configuration that the configuration statements contained in a configuration group are needed.

Configuration Groups Configuration Guidelines

For areas of your configuration to inherit configuration statements, you must first put the statements into a configuration group and then apply that group to the levels in the configuration hierarchy that require the statements. This section covers the following topics:

Creating a Configuration Group

To create a configuration group, include the **groups** statement at the **[edit]** hierarchy level:

```
[edit]
groups {
    group-name {
        configuration-data;
    }
    lccn-re0 {
        configuration-data;
    }
    lccn-re1 {
        configuration-data;
    }
}
```

group-name is the name of a configuration group. You can configure more than one configuration group by specifying multiple *group-name* statements. However, you cannot use the prefix **junos-** in a group name because it is reserved for use by the JUNOS software.

One reason for the naming restriction is a configuration group called **junos-defaults**. This preset configuration group is applied to the configuration automatically. You cannot modify or remove the **junos-defaults** configuration group. For more information about the JUNOS default configuration group, see "Using JUNOS Default Groups" on page 176.

On routers that support multiple Routing Engines, you can also specify two special group names:

- **re0**—Configuration statements applied to the Routing Engine in slot 0.
- **re1**—Configuration statements applied to the Routing Engine in slot 1.

The configuration specified in group **re0** is only applied if the current Routing Engine is in slot 0; likewise, the configuration specified in group **re1** is only applied if the current Routing Engine is in slot 1. Therefore, both Routing Engines can use the same configuration file, each using only the configuration statements that apply to it. Each **re0** or **re1** group contains at a minimum the configuration for the hostname and the management interface (**fxp0**). If each Routing Engine uses a different management interface, the group also should contain the configuration for the backup router and static routes.

In addition, the TX Matrix platform supports group names for the Routing Engines in each T640 routing node attached to the routing matrix. Providing special group names for all Routing Engines in the routing matrix allows you to configure the individual Routing Engines in each T640 routing node differently. Parameters that are not configured at the **[edit groups]** hierarchy level apply to all Routing Engines in the routing matrix.

configuration-data contains the configuration statements applied elsewhere in the configuration with the apply-groups statement. To have a configuration inherit the statements in a configuration group, include the apply-groups statement. For information about the apply-groups statement, see "Applying a Configuration Group" on page 157.

The group names for Routing Engines on the TX Matrix platform have the following formats:

- lcc*n*-re0—Configuration statements applied to the Routing Engine in slot 0 in a specified T640 routing node.
- lcc*n*-re1—Configuration statements applied to the Routing Engine in slot 1 in a specified T640 routing node.

n identifies the T640 routing node and can be from 0 through 3. For example, to configure Routing Engine 1 properties for lcc3, you include statements at the [edit groups lcc3–re1] hierarchy level. For information about the TX Matrix platform and routing matrix, see the *JUNOS System Basics Configuration Guide*.

Applying a Configuration Group

To have a configuration inherit the statements in a configuration group, include the apply-groups statement:

```
apply-groups [ group-names ];
```

If you specify more than one group name, list them in order of inheritance priority. The configuration data in the first group takes priority over the data in subsequent groups.

For routers that support multiple Routing Engines, you can specify re0 and re1 group names. The configuration specified in group re0 is only applied if the current Routing Engine is in slot 0; likewise, the configuration specified in group re1 is only applied if the current Routing Engine is in slot 1. Therefore, both Routing Engines can use the same configuration file, each using only the configuration statements that apply to it. Each re0 or re1 group contains at a minimum the configuration for the hostname and the management interface (fxp0). If each Routing Engine uses a different management interface, the group also should contain the configuration for the backup router and static routes.

You can include only one apply-groups statement at each specific level of the configuration hierarchy. The apply-groups statement at a specific hierarchy level lists the configuration groups to be added to the containing statement's list of configuration groups.

Values specified at the specific hierarchy level override values inherited from the configuration group.

Groups listed in nested apply-groups statements take priority over groups in outer statements. In the following example, the BGP neighbor 10.0.0.1 inherits configuration data from group one first, then from groups two and three. Configuration data in group one overrides data in any other group. Data from group ten is used only if a statement is not contained in any other group.

```
apply-groups [ eight nine ten ];
protocols {
    apply-groups seven;
    bgp {
        apply-groups [ five six ];
        group some-bgp-group {
            apply-groups four;
            neighbor 10.0.0.1 {
                apply-groups [ one two three ];
            }
        }
    }
}
```

Example: Configuring and Applying Configuration Groups

In this example, the Simple Network Management Protocol (SNMP) configuration is divided between the group basic and the normal configuration hierarchy.

There are a number of advantages to placing the system-specific configuration (SNMP contact) into a configuration group and thus separating it from the normal configuration hierarchy—the user can replace (using the load replace command) either section without discarding data from the other.

In addition, setting a contact for a specific box is now possible because the group data would be hidden by the router-specific data.

```
[edit]
groups {
    basic {        # User-defined group name
        snmp {     # This group contains some SNMP data
            contact "My Engineering Group";
            community BasicAccess {
                authorization read-only;
            }
        }
    }
}
apply-groups basic;    # Enable inheritance from group "basic"
snmp {                 # Some normal (non-group) configuration
    location "West of Nowhere";
}
```

This configuration is equivalent to the following:

```
[edit]
snmp {
    location "West of Nowhere";
    contact "My Engineering Group";
    community BasicAccess {
        authorization read-only;
    }
}
```

For information about how to disable inheritance of a configuration group, see "Disabling Inheritance of a Configuration Group" on page 160.

Example: Creating and Applying Configuration Groups on a TX Matrix Platform

```
[edit]
groups {
    re0 { # Routing Engine 0 on TX Matrix platform
        system {
            host-name <host-name>;
            backup-router <ip-address>;
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address <ip-address>;
                    }
                }
            }
        }
    }
    re1 { # Routing Engine 1 on TX Matrix platform
        system {
            host-name <host-name>;
            backup-router <ip-address>;
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address <ip-address>;
                    }
                }
            }
        }
    }
```

```
    lcc0-re0 { # Routing Engine 0 on T640 routing node numbered 0
      system {
        host-name <host-name>;
        backup-router <ip-address>;
      }
      interfaces {
        fxp0 {
          unit 0 {
          family inet {
          address <ip-address>;
        }
      }
    }
    lcc0-re1 { # Routing Engine 1 on T640 routing node numbered 0
      system {
      host-name <host-name>;
      backup-router <ip-address>;
    }
      interfaces {
        fxp0 {
          unit 0 {
            family inet {
            address <ip-address>;
            }
        }
    }
}
apply-groups [ re0 re1 lcc0-re0 lcc0-re1 ];
```

Disabling Inheritance of a Configuration Group

To disable inheritance of a configuration group at any level except the top level of the hierarchy, include the apply-groups-except statement:

```
apply-groups-except [ group-names ];
```

This useful when you use the apply-group statement at a specific hierarchy level but also want to override the values inherited from the configuration group for a specific parameter.

Example: Disabling Inheritance on Interface s0-1/1/0

In the following example, the apply-groups statement is applied globally at the interfaces level. The apply-groups-except statement is also applied at interface s0-1/1/0 so that it uses the default values hold-time and link-mode.

```
[edit]
groups {              # "groups" is a top-level statement
   global {           # User-defined group name
      interfaces {
         <*> {
         hold-time down 640;
         link-mode full-duplex;
      }
   }
}
```

```
apply-groups global;
interfaces {
    so-1/1/0 {
      apply-groups-except global;  # Disables inheritance from group "global":
    }                              # so-1/1/0 uses default values for “hold-time”
                                   # and “link-mode”
}
```

For information about applying a configuration group, see “Applying a Configuration Group” on page 157.

Displaying Inherited Values

Configuration groups can add some confusion regarding the actual values used by the router, because configuration data can be inherited from configuration groups. To view the actual values used by the router, use the display inheritance command after the pipe in a show command. This command displays the inherited statements at the level at which they are inherited and the group from which they have been inherited.

```
[edit]
user@host# show | display inheritance
snmp {
    location "West of Nowhere";
    ##
    ## ’My Engineering Group’ was inherited from group ’basic’
    ##
    contact "My Engineering Group";
    ##
    ## ’BasicAccess’ was inherited from group ’basic’
    ##
    community BasicAccess {
      ##
      ## ’read-only’ was inherited from group ’basic’
      ##
      authorization read-only;
    }
}
```

To display the expanded configuration (the configuration, including the inherited statements) without the ## lines, use the except command after the pipe in a show command:

```
[edit]
user@host# show | display inheritance | except ##
snmp {
    location "West of Nowhere";
    contact "My Engineering Group";
    community BasicAccess {
      authorization read-only;
    }
}
```

Using Wildcards with Configuration Groups

You can use wildcards to identify names and allow one statement to provide data for a variety of statements. For example, grouping the configuration of the sonet-options statement over all SONET/SDH interfaces or the dead interval for Open Shortest Path First (OSPF) over all Asynchronous Transfer Mode (ATM) interfaces simplifies configuration files and eases their maintenance.

Wildcarding in normal configuration data is done in a style that is consistent with traditional UNIX shell name wildcarding. In this style of wildcarding, you can use the following metacharacters:

- Asterisk (*)—Matches any string of characters.
- Question mark (?)—Matches any single character.
- Open bracket ([ )—Introduces a character class.
- Close bracket ( ])—Indicates the end of a character class. If the close bracket is missing, the open bracket matches a [rather than introduce a character class.
- A character class matches any of the characters between the square brackets. Character classes must be enclosed in quotation marks (“ ”).
- Hyphen (-)—Specifies a range of characters.
- Exclamation point (!)—The character class can be complemented by making an exclamation point the first character of the character class. To include a] in a character class, make it the first character listed (after the !, if any). To include a minus sign, make it the first or last character listed.

Wildcarding in configuration groups follows the same rules, but the wildcard pattern must be enclosed in angle brackets (<*pattern*>) to differentiate it from other wildcarding in the configuration file. For example:

```
[edit]
groups {
    sonet-default {
        interfaces {
            <so-*> {
                sonet-options {
                    payload-scrambler;
                    rfc-2615;
                }
            }
        }
    }
}
```

Wildcard expressions match (and provide configuration data for) existing statements in the configuration that match their expression only. In the example above, the expression <so-*> passes its sonet-options statement to any interface that matches the expression so-*.

Angle brackets allow you to pass normal wildcarding through without modification. In all matching within the configuration, whether it is done with or without wildcards, the first item encountered in the configuration that matches is used. In the following example, data from the wildcarded BGP groups is inherited in the order in which the groups are listed. The preference value from <*a*> overrides the preference in <*b*>, just as the p value from <*c*> overrides the one from <*d*>. Data values from any of these groups override the data values from abcd.

```
[edit]
user@host# show
groups {
    one {
        protocols {
            bgp {
                group <*a*> {
                    preference 1;
                }
                group <*b*> {
                    preference 2;
                }
                group <*c*> {
                    out-delay 3;
                }
                group <*d*> {
                    out-delay 4;
                }
                group abcd {
                    preference 10;
                    hold-time 10;
                    out-delay 10;
                }
            }
        }
    }
}
protocols {
    bgp {
        group abcd {
            apply-groups one;
        }
    }
}
```

```
[edit]
user@host# show | display inheritance
protocols {
    bgp {
        group abcd {
            ##
            ## '1' was inherited from group 'one'
            ##
            preference 1;
            ##
            ## '10' was inherited from group 'one'
            ##
            hold-time 10;
            ##
            ## '3' was inherited from group 'one'
            ##
            out-delay 3;
        }
    }
}
```

Example: Using Wildcards with Configuration Groups

The following example demonstrates the use of wildcarding. The interface so-0/0/0 inherits data from the various SONET/SDH interface wildcard patterns in group one.

```
[edit]
user@host# show
groups {
    one {
        interfaces {
            <so-*> {
                sonet-options {
                    rfc-2615;
                }
            }
            <so-0/*> {
                sonet-options {
                    fcs 32;
                }
            }
            <so-*/0/*> {
                sonet-options {
                    fcs 16;
                }
            }
            <so-*/*/0> {
                sonet-options {
                    payload-scrambler;
                }
            }
        }
    }
}
```

```
apply-groups one;
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.0.0.1/8;
            }
        }
    }
}
[edit]
user@host# show | display inheritance
interfaces {
    so-0/0/0 {
        ##
        ## 'sonet-options' was inherited from group 'one'
        ##
        sonet-options {
            ##
            ## '32' was inherited from group 'one'
            ##
            fcs 32;
            ##
            ## 'payload-scrambler' was inherited from group 'one'
            ##
            payload-scrambler;
            ##
            ## 'rfc-2615' was inherited from group 'one'
            ##
            rfc-2615;
        }
        unit 0 {
            family inet {
                address 10.0.0.1/8;
            }
        }
    }
}
```

Examples: Configuration Groups

The following examples illustrate ways to use configuration groups and inheritance:

- Configuring Sets of Statements with Configuration Groups on page 166
- Configuring Interfaces on page 167
- Configuring a Consistent Management IP Address on page 169

Configuring Sets of Statements with Configuration Groups

When sets of statements exist in configuration groups, all values are inherited. For example:

```
[edit]
user@host# show
groups {
    basic {
        snmp {
            interface so-1/1/1.0;
        }
    }
}
apply-groups basic;
snmp {
    interface so-0/0/0.0;
}
[edit]
user@host# show | display inheritance
snmp {
    ##
    ## 'so-1/1/1.0' was inherited from group 'basic'
    ##
    interface [ so-0/0/0.0 so-1/1/1.0 ];
}
```

For sets that are not displayed within brackets, all values are also inherited. For example:

```
[edit]
user@host# show
groups {
    worldwide {
        system {
            name-server {
                10.0.0.100;
                10.0.0.200;
            }
        }
    }
}
apply-groups worldwide;
system {
    name-server {
        10.0.0.1;
        10.0.0.2;
    }
}
```

```
[edit]
user@host# show | display inheritance
system {
    name-server {
    ##
    ## '10.0.0.100' was inherited from group 'worldwide'
    ##
      10.0.0.100;
    ##
    ## '10.0.0.200' was inherited from group 'worldwide'
    ##
      10.0.0.200;
    }
}
```

Configuring Interfaces

You can use configuration groups to separate the common interface media parameters from the interface-specific addressing information. The following example places configuration data for ATM interfaces into a group called atm-options:

```
[edit]
user@host# show
groups {
    atm-options {
      interfaces {
        <at-*> {
          atm-options {
            vpi 0 maximum-vcs 1024;
          }
          unit <*> {
            encapsulation atm-snap;
            point-to-point;
            family iso;
          }
        }
      }
    }
}
apply-groups atm-options;
interfaces {
    at-0/0/0 {
      unit 100 {
        vci 0.100;
        family inet {
          address 10.0.0.100/30;
        }
      }
      unit 200 {
        vci 0.200;
        family inet {
          address 10.0.0.200/30;
        }
      }
    }
}
```

```
[edit]
user@host# show | display inheritance
interfaces {
    at-0/0/0 {
        ##
        ## "atm-options" was inherited from group "atm-options"
        ##
        atm-options {
            ##
            ## "1024" was inherited from group "atm-options"
            ##
            vpi 0 maximum-vcs 1024;
        }
        unit 100 {
            ##
            ## "atm-snap" was inherited from group "atm-options"
            ##
            encapsulation atm-snap;
            ##
            ## "point-to-point" was inherited from group "atm-options"
            ##
            point-to-point;
            vci 0.100;
            family inet {
                address 10.0.0.100/30;
            }
            ##
            ## "iso" was inherited from group "atm-options"
            ##
            family iso;
        }
        unit 200 {
            ##
            ## "atm-snap" was inherited from group "atm-options"
            ##
            encapsulation atm-snap;
            ##
            ## "point-to-point" was inherited from group "atm-options"
            ##
            point-to-point;
            vci 0.200;
            family inet {
                address 10.0.0.200/30;
            }
            ##
            ## "iso" was inherited from group "atm-options"
            ##
            family iso;
        }
    }
}
```

```
[edit]
user@host# show | display inheritance | except ##
interfaces {
    at-0/0/0 {
        atm-options {
            vpi 0 maximum-vcs 1024;
        }
        unit 100 {
            encapsulation atm-snap;
            point-to-point;
            vci 0.100;
            family inet {
                address 10.0.0.100/30;
            }
            family iso;
        }
        unit 200 {
            encapsulation atm-snap;
            point-to-point;
            vci 0.200;
            family inet {
                address 10.0.0.200/30;
            }
            family iso;
        }
    }
}
```

Configuring a Consistent Management IP Address

On platforms with multiple Routing Engines, each Routing Engine is configured with a separate IP address for the management interface (fxp0). To access the master Routing Engine, you must know which Routing Engine is active and use the appropriate IP address.

Optionally, for consistent access to the master Routing Engine, you can configure an additional IP address and use this address for the management interface regardless of which Routing Engine is active. This additional IP address is active only on the management interface for the master Routing Engine. During switchover, the address moves to the new master Routing Engine.

In the following example, address 10.17.40.131 is configured for both Routing Engines and includes a master-only statement. With this configuration, the 10.17.40.131 address is active only on the master Routing Engine. The address remains consistent regardless of which Routing Engine is active. Address 10.17.40.132 is assigned to fxp0 on re0, and 10.17.40.133 is assigned to fxp0 on re1.

```
[edit groups re0 interfaces fxp0]
unit 0 {
    family inet {
      address 10.17.40.131/25 {
            master-only;
       }
      address 10.17.40.132/25;
    }
}

[edit groups re1 interfaces fxp0]
unit 0 {
    family inet {
      address 10.17.40.131/25 {
            master-only;
       }
      address 10.17.40.133/25;
    }
}
```

This feature is available on all platforms that include dual Routing Engines. On the TX Matrix platform, this feature is applicable to the switch-card chassis (SCC) only.

Configuring Peer Entities

In this example, we create a group **some-isp** that contains configuration data relating to another Internet service provider (ISP). We can then insert **apply-group** statements at any point to allow any location in the configuration hierarchy to inherit this data.

```
[edit]
user@host# show
groups {
    some-isp {
        interfaces {
            <ge-*> {
                gigether-options {
                    flow-control;
                }
            }
        }
        protocols {
            bgp {
                group <*> {
                    neighbor <*> {
                        remove-private;
                    }
                }
            }
            pim {
                interface <*> {
                    version 1;
                }
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        apply-groups some-isp;
        unit 0 {
            family inet {
                address 10.0.0.1/24;
            }
        }
    }
}
protocols {
    bgp {
        group main {
            neighbor 10.254.0.1 {
                apply-groups some-isp;
            }
        }
    }
```

```
        pim {
            interface ge-0/0/0.0 {
                apply-groups some-isp;
            }
        }
}
[edit]
user@host# show | display inheritance
interfaces {
    ge-0/0/0 {
        ##
        ## "gigether-options" was inherited from group "some-isp"
        ##
        gigether-options {
            ##
            ## "flow-control" was inherited from group "some-isp"
            ##
            flow-control;
        }
        unit 0 {
            family inet {
                address 10.0.0.1/24;
            }
        }
    }
}
protocols {
    bgp {
        group main {
            neighbor 10.254.0.1 {
                ##
                ## "remove-private" was inherited from group "some-isp"
                ##
                remove-private;
            }
        }
    }
    pim {
        interface ge-0/0/0.0 {
            ##
            ## "1" was inherited from group "some-isp"
            ##
            version 1;
        }
    }
}
```

Establishing Regional Configurations

In this example, one group is populated with configuration data that is standard throughout the company, while another group contains regional deviations from this standard:

```
[edit]
user@host# show
groups {
    standard {
        interfaces {
            <t3-*> {
                t3-options {
                    compatibility-mode larscom subrate 10;
                    idle-cycle-flag ones;
                }
            }
        }
    }
    northwest {
        interfaces {
            <t3-*> {
                t3-options {
                    long-buildout;
                    compatibility-mode kentrox;
                }
            }
        }
    }
}
apply-groups standard;
interfaces {
    t3-0/0/0 {
        apply-groups northwest;
    }
}
[edit]
user@host# show | display inheritance
interfaces {
    t3-0/0/0 {
        ##
        ## "t3-options" was inherited from group "northwest"
        ##
        t3-options {
            ##
            ## "long-buildout" was inherited from group "northwest"
            ##
            long-buildout;
            ##
```

```
            ## "kentrox" was inherited from group "northwest"
            ##
            compatibility-mode kentrox;
            ##
            ## "ones" was inherited from group "standard"
            ##
            idle-cycle-flag ones;
        }
    }
}
```

Selecting Wildcard Names

You can combine wildcarding and thoughtful use of names in statements to tailor statement values:

```
[edit]
user@host# show
groups {
    mpls-conf {
        protocols {
            mpls {
                label-switched-path <*-major> {
                    retry-timer 5;
                    bandwidth 155m;
                    optimize-timer 60;
                }
                label-switched-path <*-minor> {
                    retry-timer 15;
                    bandwidth 64k;
                    optimize-timer 120;
                }
            }
        }
    }
}
apply-groups mpls-conf;
protocols {
    mpls {
        label-switched-path metro-major {
            to 10.0.0.10;
        }
        label-switched-path remote-minor {
            to 10.0.0.20;
        }
    }
}
```

```
[edit]
user@host# show | display inheritance
protocols {
    mpls {
        label-switched-path metro-major {
            to 10.0.0.10;
            ##
            ## "5" was inherited from group "mpls-conf"
            ##
            retry-timer 5;
            #
            ## "155m" was inherited from group "mpls-conf"
            ##
            bandwidth 155m;
            ##
            ## "60" was inherited from group "mpls-conf"
            ##
            optimize-timer 60;
        }
        label-switched-path remote-minor {
            to 10.0.0.20;
            ##
            ## "15" was inherited from group "mpls-conf"
            ##
            retry-timer 15;
            ##
            ## "64k" was inherited from group "mpls-conf"
            ##
            bandwidth 64k;
            ##
            ## "120" was inherited from group "mpls-conf"
            ##
            optimize-timer 120;
        }
    }
}
```

Using JUNOS Default Groups

The JUNOS software provides a hidden and immutable configuration group called junos-defaults that is automatically applied to the configuration of your routing platform. The junos-defaults group contains preconfigured statements that contain predefined values for common applications. Some of the statements must be referenced to take effect, such as definitions for applications (for example, FTP or telnet settings). Other statements are applied automatically, such as terminal settings.

NOTE: Many identifiers included in the junos-defaults configuration group begin with the name junos-. Because identifiers beginning with the name junos- are reserved for use by Juniper Networks, you cannot define any configuration objects using this name.

You cannot include junos-defaults as a configuration group name in an apply-groups statement.

To view the full set of available preset statements from the JUNOS default group, issue the show groups junos-defaults configuration mode command at the top level of the configuration. The following example displays a partial list of JUNOS default groups:

```
user@host# show groups junos-defaults
#
# Make vt100 the default for the console port
#
system {
    ports {
        console type vt100;
    }
}
applications {
    #
    # File Transfer Protocol
    #
    application junos-ftp {
        application-protocol ftp;
        protocol tcp;
        destination-port 21;
    }
    #
    # Trivial File Transfer Protocol
    #
    application junos-tftp {
        application-protocol tftp;
        protocol udp;
        destination-port 69;
    }
    #
    # RPC port mapper on TCP
    #
```

```
        application junos-rpc-portmap-tcp {
            application-protocol rpc-portmap;
            protocol tcp;
            destination-port 111;
        }
        #
        # RPC port mapper on UDP
        #
        }
}
```

To reference statements available from the junos-defaults group, include the selected junos- *default-name* statement at the applicable hierarchy level.

Example: Referencing the Preset Statement

The following example is a preset statement from the JUNOS defaults group that is available for FTP in a stateful firewall:

```
[edit]
groups {
    junos-defaults {
        applications {
            application junos-ftp {              # Use FTP default configuration
                application-protocol ftp;
                protocol tcp;
                destination-port 21;
            }
        }
    }
}
```

To reference a preset JUNOS default statement from the JUNOS defaults group, include the junos- *default-name* statement at the applicable hierarchy level. For example, to reference the JUNOS default statement for FTP in a stateful firewall, include the junos-ftp statement at the [edit services stateful-firewall rule *rule-name* term *term-name* from applications] hierarchy level:

```
[edit]
services {
    stateful-firewall {
        rule my-rule {
            term my-term {
                from {
                    applications junos-ftp; #Reference predefined statement, junos-ftp,
                }                           #for FTP in the stateful firewall configuration
            }
        }
    }
}
```

Example: Viewing Default Statements That Have Been Applied to the Configuration

To view the JUNOS defaults that have been applied to the configuration, issue the show | display inheritance defaults command. For example, to view the inherited JUNOS defaults at the [edit system ports] hierarchy level:

```
user@host# show system ports | display inheritance defaults
## ## 'console' was inherited from group 'junos-defaults'
## 'vt100' was inherited from group 'junos-defaults'
## console type vt100;
```

If you choose not to use existing JUNOS default statements, you can create your own configuration groups manually. For more information about manually creating of configuration groups, see “Overview” on page 154 and “Configuration Groups Configuration Statements” on page 155.

Chapter 12

Summary of Configuration Group Statements

The following sections explain each of the configuration group statements. The statements are organized alphabetically.

apply-groups

Syntax apply-groups [*group-names*];

Hierarchy Level All hierarchy levels

Release Information Statement introduced before JUNOS Release 7.4.

Description Apply a configuration group to a specific hierarchy level in a configuration, to have a configuration inherit the statements in the configuration group.

You can specify more than one group name. You must list them in order of inheritance priority. The configuration data in the first group takes priority over the data in subsequent groups.

For routers that support multiple Routing Engines, you can specify re0 and re1 as group names. The configuration specified in group re0 is applied only if the current Routing Engine is in slot 0; likewise, the configuration specified in group re1 is applied only if the current Routing Engine is in slot 1. Therefore, both Routing Engines can use the same configuration file, each using only the configuration statements that apply to it. Each re0 or re1 group contains at a minimum the configuration for the hostname and the management interface (fxp0). If each Routing Engine uses a different management interface, the group also should contain the configuration for the backup router and static routes.

For the TX Matrix platform, you can specify lcc*n*-re0 and lcc*n*-re1 as group names where *n* identifies the T640 routing node that is connected to a TX Matrix platform and can be from 0 through 3.

You can include the apply-groups statement at any level of the configuration hierarchy.

You can include only one apply-groups statement at each specific level of the configuration hierarchy. The apply-groups statement at a specific hierarchy level lists the configuration groups to be added to the containing statement's list of configuration groups.

Options *group-name*—One or more names specified in the groups statement.

Usage Guidelines See "Applying a Configuration Group" on page 157.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

See Also groups on page 180

apply-groups-except

Syntax apply-groups-except [*group-names*];

Hierarchy Level All hierarchy levels except the top level

Description Disables inheritance of a configuration group.

Options *group-names*—One or more names specified in the groups statement.

Usage Guidelines See "Disabling Inheritance of a Configuration Group" on page 160.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

See Also groups on page 180

groups

Syntax
```
groups {
    group-name {
      configuration-data;
    }
    lccn-re0 {
      configuration-data;
    }
    lccn-re1 {
      configuration-data;
    }
}
```

Hierarchy Level [edit]

Release Information Statement introduced before JUNOS Release 7.4.

Description Create a configuration group.

Options *configuration-data*—The configuration statements that are to be applied elsewhere in the configuration with the apply-groups statement, to have the target configuration inherit the statements in the group.

group-name—Name of the configuration group. To configure multiple groups, specify more than one *group-name*. On routers that support multiple Routing Engines, you can also specify two special group names:

- re0—Configuration statements that are to be applied to the Routing Engine in slot 0.
- re1—Configuration statements that are to be applied to the Routing Engine in slot 1.

The configuration specified in group re0 is applied only if the current Routing Engine is in slot 0; likewise, the configuration specified in group re1 is applied only if the current Routing Engine is in slot 1. Therefore, both Routing Engines can use the same configuration file, each using only the configuration statements that apply to it. Each re0 or re1 group contains at a minimum the configuration for the hostname and the management interface (fxp0). If each Routing Engine uses a different management interface, the group also should contain the configuration for the backup router and static routes.

(Routing matrix only) The TX Matrix platform supports group names for the Routing Engines in each connected T640 routing node in the following formats:

- lcc*n*-re0—Configuration statements applied to the Routing Engine in slot 0 of the specified T640 routing node that is connected to a TX Matrix platform.
- lcc*n*-re1—Configuration statements applied to the specified to the Routing Engine in slot 1 of the specified T640 routing node that is connected to a TX Matrix platform.

n identifies the T640 routing node and can be from 0 through 3.

Usage Guidelines See "Creating a Configuration Group" on page 156.

Required Privilege Level configure—To enter configuration mode.

See Also apply-groups on page 179, apply-groups-except on page 180

Part 4
CLI Command Summaries

Chapter 13

Summary of CLI Environment Commands

The following sections explain each of the command-line interface (CLI) environment commands described in this book. The commands are organized alphabetically.

set cli complete-on-space

Syntax set cli complete-on-space (off | on)

Release Information Command introduced before JUNOS Release 7.4.

Description Set the CLI to complete a partial command entry when you type a space or a tab. This is the default behavior of the CLI.

Options off—Turn off command completion.

on—Allow either a space or a tab to be used for command completion.

Sample Output In the following example, pressing the spacebar changes the partial command entry from com to complete-on-space. The example shows how adding the keyword off at the end of the command disables command completion.

```
user@host> set cli com<Space>
user@host>set cli complete-on-space off
Disabling complete-on-space
```

Usage Guidelines "Setting Command Completion" on page 139

Required Privilege Level view

set cli directory

Syntax set cli directory *directory*

Release Information Command introduced before JUNOS Release 7.4.

Description Set the current working directory.

Options *directory*—Pathname of the working directory.

Sample Output
```
user@host> set cli directory /var/home/regress
Current directory: /var/home/regress
```

Usage Guidelines "Setting the CLI Directory" on page 138

Required Privilege Level view

set cli idle-timeout

Syntax set cli idle-timeout
<*minutes*>

Release Information Command introduced before JUNOS Release 7.4.

Description Set the maximum time that an individual session can be idle before the user is logged off the router.

Options *minutes*—(Optional) Maximum idle time. The range of values, in minutes, is 0 through 100,000. If you do not issue this command, and the user's login class does not specify this value, the user is never forced off the system after extended idle times. Setting the value to 0 disables the timeout.

Sample Output
```
user@host> set cli idle-timeout 60
Idle timeout set to 60 minutes
```

Usage Guidelines "Setting the Idle Timeout" on page 139

Required Privilege Level view

set cli prompt

Syntax set cli prompt *string*

Release Information Command introduced before JUNOS Release 7.4.

Description Set the prompt so that it is displayed within the CLI.

Options *string*—CLI prompt string. To include spaces in the prompt, enclose the string in quotation marks. By default, the string is *username@hostname*.

Sample Output
```
user@host> set cli prompt lab1-router>
lab1-router>
```

Usage Guidelines See "Setting the CLI Prompt" on page 138.

Required Privilege Level view

set cli restart-on-upgrade

Syntax set cli restart-on-upgrade string (off | on)

Release Information Command introduced before JUNOS Release 7.4.

Description For an individual session, set the CLI to prompt you to restart the router after upgrading the software.

Options off—Disables the prompt.

on—Enables the prompt.

Sample Output
```
user@host> set cli restart-on-upgrade on
Enabling restart-on-upgrade
```

Usage Guidelines See "Setting the CLI to Prompt After a Software Upgrade" on page 139.

Required Privilege Level view

set cli screen-length

Syntax set cli screen-length *length*

Release Information Command introduced before JUNOS Release 7.4.

Description Set terminal screen length.

Options *length*—Number of lines of text that the terminal screen displays. The range of values, in number of lines, is 24 through 100,000. The default is 24.

The point at which the —(more)— prompt appears on the screen is a function of this setting and the settings for the set cli screen-width and set cli terminal commands.

See Also set cli screen-width on page 188
set cli terminal on page 189
show cli on page 190

Sample Output
```
user@host> set cli screen-length 75
Screen length set to 75
```

Usage Guidelines See "Setting the Screen Length" on page 138.

Required Privilege Level view

set cli screen-width

Syntax set cli screen-width *width*

Release Information Command introduced before JUNOS Release 7.4.

Description Set the terminal screen width.

Options *width*—Number of characters in a line. The range of values is 80 through 100,000. The default is 80.

The point at which the —(more)— prompt appears on the screen is a function of this setting and the settings for the set cli screen-length and set cli terminal commands.

See Also set cli screen-length on page 188
set cli terminal on page 189
show cli on page 190

Sample Output
```
user@host> set cli screen-width
Screen width set to 132
```

Usage Guidelines See "Setting the Screen Width" on page 138.

Required Privilege Level view

set cli terminal

Syntax set cli terminal *terminal-type*

Release Information Command introduced before JUNOS Release 7.4.

Description Set the terminal type.

Options *terminal-type*—Type of terminal that is connected to the Ethernet management port:

- ansi—ANSI-compatible terminal (80 characters by 24 lines)
- small-xterm—Small xterm window (80 characters by 24 lines)
- vt100—VT100-compatible terminal (80 characters by 24 lines)
- xterm—Large xterm window (80 characters by 65 lines)

Sample Output

```
user@host> set cli terminal xterm
```

Usage Guidelines See "Setting the Terminal Type" on page 138.

Required Privilege Level view

set cli timestamp

Syntax set cli timestamp (format *timestamp-format* | disable)

Release Information Command introduced before JUNOS Release 7.4.

Description Set a timestamp for CLI output.

Options format *timestamp-format*—Set the data and time format for the timestamp. The timestamp format you specify can include the following placeholders in any order:

- %m—Two-digit month
- %d—Two-digit date
- %T—Six-digit hour, minute, and seconds

Enclose the format in single quotation marks (‘). Do not use spaces. Use a hyphen (-) or similar character to separate placeholders.

disable—Remove the timestamp from the CLI.

Sample Output

```
user@host> set cli timestamp format '%m-%d-%T'
'04-21-17:39:13'
CLI timestamp set to: '%m-%d-%T'
```

Usage Guidelines See "Setting the CLI Timestamp" on page 139.

Required Privilege Level view

set date

Syntax set date (*date-time* | ntp <*ntp-server*> <source-address *source-address*>)

Release Information Command introduced before JUNOS Release 7.4.

Description Set the data and time.

Options *date-time*—Date and time. Enter this string inside quotation marks.

ntp—Use a Network Time Protocol (NTP) server to synchronize the current date and time setting on the router.

ntp-server—(Optional) Specify the IP address of one or more NTP servers.

source-address *source-address*—(Optional) Specify the source address that is used by the router to contact the remote NTP server.

Sample Output
```
user@host> set date ntp
21 Apr 17:22:02 ntpdate[3867]: step time server 172.17.27.46 offset 8.759252 sec
```

Required Privilege Level view

show cli

Syntax show cli

Release Information Command introduced before JUNOS Release 7.4.

Description Display configured CLI settings.

Sample Output
```
user@host> show cli
CLI complete-on-space set to on
CLI idle-timeout disabled
CLI restart-on-upgrade set to on
CLI screen-length set to 47
CLI screen-width set to 132
CLI terminal is 'vt100'
CLI is operating in enhanced mode
CLI timestamp disabled
CLI working directory is '/var/home/regress'
```

Required Privilege Level view

show cli authorization

Syntax show cli authorization

Release Information Command introduced before JUNOS Release 7.4.

Description Display the permissions for the current user.

Sample Output

```
user@host> show cli authorization
Current user: 'root' login: ‘boojum’ class '(root)'
Permissions:
    admin               -- Can view user accounts
    admin-control       -- Can modify user accounts
    clear               -- Can clear learned network information
    configure           -- Can enter configuration mode
    control             -- Can modify any config
    edit                -- Can edit full files
    field               -- Special for field (debug) support
    floppy              -- Can read and write from the floppy
    interface           -- Can view interface config
    interface-control   -- Can modify interface config
    network             -- Can access the network
    reset               -- Can reset/restart interfaces and daemons
    routing             -- Can view routing config
    routing-control     -- Can modify routing config
    shell               -- Can start a local shell
    snmp                -- Can view SNMP config
    snmp-control        -- Can modify SNMP config
    system              -- Can view system config
    system-control      -- Can modify system config
    trace               -- Can view trace file settings
    trace-control       -- Can modify trace file settings
    view                -- Can view current values and statistics
    maintenance         -- Can become the super-user
    firewall            -- Can view firewall config
    firewall-control    -- Can modify firewall config
    secret              -- Can view secret config
    secret-control      -- Can modify secret config
```

Required Privilege Level view

show cli directory

Syntax show cli directory

Release Information Command introduced before JUNOS Release 7.4.

Description Display the current working directory.

Sample Output

```
user@host> show cli directory
Current directory: /var/home/regress
```

Required Privilege Level view

show cli history

Syntax show cli history
<*count*>

Release Information Command introduced before JUNOS Release 7.4.

Description Display a list of previous CLI commands.

Options none—Display all previous CLI commands.

count—(Optional) Maximum number of commands to display.

Sample Output

```
user@host> show cli history
  11:14:14 -- show arp
  11:22:10 -- show cli authorization
  11:27:12 -- show cli history
```

Usage Guidelines See “Displaying CLI Command History” on page 32.

Required Privilege Level view

Chapter 14

Summary of CLI Configuration Mode Commands

The following sections explain each of the command-line interface (CLI) configuration mode commands describes in this book. The commands are organized alphabetically.

activate

Syntax activate (*statement* | *identifier*)

Release Information Command introduced before JUNOS Release 7.4.

Description Remove the inactive: tag from a statement, effectively adding the statement or identifier back to the configuration. Statements or identifiers that have been activated take effect when you next issue the commit command.

Options *identifier*—Identifier from which you are removing the inactive tag. It must be an identifier at the current hierarchy level.

statement—Statement from which you are removing the inactive tag. It must be a statement at the current hierarchy level.

Usage Guidelines See "Deactivating and Reactivating Statements and Identifiers" on page 88.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

See Also deactivate on page 197

annotate

Syntax annotate *statement* "*comment-string*"

Release Information Command introduced before JUNOS Release 7.4.

Description Add comments to a configuration. You can add comments only at the current hierarchy level.

Any comments you add appear only when you view the configuration by entering the show command in configuration mode or the show configuration command in operational mode.

Options *comment-string*—Text of the comment. You must enclose it in quotation marks. In the comment string, you can include the comment delimiters /* */ or #. If you do not specify any, the comment string is enclosed with the /* */ comment delimiters. If a comment for the specified *statement* already exists, it is deleted and replaced with the new comment.

statement—Statement to which you are attaching the comment.

Usage Guidelines See "Adding Comments in a Configuration" on page 89.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

See Also See the description statement in the *JUNOS Network Interfaces Configuration Guide*.

commit

Syntax commit <<*at* <"string">> <and-quit> <check> <comment <"*comment-string*">> <confirmed> <display detail> <*minutes*> <synchronize>

Release Information Command introduced before JUNOS Release 7.4.

Description Commit the set of changes to the database and cause the changes to take operational effect.

Options at <"string">—(Optional) Save software configuration changes and activate the configuration at a future time, or upon reboot.

string is reboot or the future time to activate the configuration changes. Enclose the *string* value (including reboot) in quotation marks ("). You can specify time in two formats:

- A time value in the form *hh*:*mm* [:ss] (hours, minutes, and optionally seconds)— Commit the configuration at the specified time, which must be in the future but before 11:59:59 PM on the day the commit at configuration command is issued. Use 24-hour time for the *hh* value; for example, 04:30:00 is 4:30:00 AM, and 20:00 is 8:00 PM. The time is interpreted with respect to the clock and time zone settings on the router.

- A date and time value in the form *yyyy-mm-dd hh*:*mm* [:ss] (year, month, date, hours, minutes, and, optionally, seconds)—Commit the configuration at the specified day and time, which must be after the commit at command is issued. Use 24-hour time for the *hh* value. For example, 2003-08-21 12:30:00 is 12:30 PM on August 21, 2003. The time is interpreted with respect to the clock and time zone settings on the router.

For example, commit at "18:00:00". For date and time, include both values in the same set of quotation marks. For example, commit at "2005-03-10 14:00:00".

A *commit check* is performed when you issue the commit at configuration mode command. If the result of the check is successful, then the current user is logged out of configuration mode, and the configuration data is left in a read-only state. No other commit can be performed until the scheduled commit is completed.

NOTE: If the JUNOS software fails before the configuration changes become active, all configuration changes are lost.

You cannot issue the commit at configuration command when there is a pending reboot.

You cannot issue the request system reboot command once you schedule a commit operation for a specific time in the future.

You cannot commit a configuration when a scheduled commit is pending. For information about how to use the clear command to cancel a scheduled configuration, see the *JUNOS System Basics and Services Command Reference*.

and-quit—(Optional) Commit the configuration and, if the configuration contains no errors and the commit succeeds, exit from configuration mode.

check—(Optional) Verify the syntax of the configuration, but do not activate it.

comment <"*comment-string*"> —(Optional) Add a comment that describes the committed configuration. The comment can be as long as 512 bytes and must be typed on a single line. You cannot include a comment with the commit check command. Enclose *comment-string* in quotation marks ("). For example, commit comment "Includes changes recommended by SW Lab".

confirmed <*minutes*>—(Optional) Require that the commit be confirmed within the specified amount of time. To confirm a commit, enter either a commit or commit check command. If the commit is not confirmed within the time limit, the configuration rolls back automatically to the precommit configuration and a broadcast message is sent to all logged-in users. To show when a rollback is scheduled, enter the show system commit command.
Range: 1 through 65,535 minutes
Default: 10 minutes

display detail—(Optional) Monitors the commit process.

synchronize—(Optional) If your router has two Routing Engines, you can manually direct one Routing Engine to synchronize its configuration with the other by issuing the commit synchronize command. The Routing Engine on which you execute this command (request Routing Engine) copies and loads its candidate configuration to the other (responding Routing Engine). Both Routing Engines then perform a syntax check on the candidate configuration file being committed. If no errors are found, the configuration is activated and becomes the current operational configuration on both Routing Engines. The commit synchronize command does not work if the responding Routing Engine has uncommitted configuration changes.

NOTE: When you issue the commit synchronize command, you must use the apply-groups re0 and re1 commands. For information about how to use groups, see "Applying a Configuration Group" on page 157.

The responding Routing Engine must use JUNOS Release 5.0 or later.

Usage Guidelines See "Verifying a Configuration" on page 92, "Committing a Configuration" on page 92, "Scheduling a Commit Operation" on page 95, "Deactivating and Reactivating Statements and Identifiers" on page 88, "Monitoring the Commit Process" on page 96, and "Adding a Comment to Describe the Committed Configuration" on page 97.

Required Privilege Level configure—To enter configuration mode.

copy

Syntax copy *existing-statement* to *new-statement*

Release Information Command introduced before JUNOS Release 7.4.

Description Make a copy of an existing statement in the configuration.

Options *existing-statement*—Statement to copy.

new-statement—Copy of the statement.

Usage Guidelines See "Copying a Statement in the Configuration" on page 84.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

deactivate

Syntax deactivate (*statement* | *identifier*)

Release Information Command introduced before JUNOS Release 7.4.

Description Add the inactive: tag to a statement, effectively commenting out the statement or identifier from the configuration. Statements or identifiers marked as inactive do not take effect when you issue the commit command.

Options *identifier*—Identifier to which you are adding the inactive: tag. It must be an identifier at the current hierarchy level.

statement—Statement to which you are adding the inactive: tag. It must be a statement at the current hierarchy level.

Usage Guidelines See "Deactivating and Reactivating Statements and Identifiers" on page 88.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

See Also activate on page 193, delete on page 197

delete

Syntax delete <*statement-path*> <*identifier*>

Release Information Command introduced before JUNOS Release 7.4.

Description Delete a statement or identifier. All subordinate statements and identifiers contained within the specified statement path are deleted with it.

Deleting a statement or an identifier effectively "unconfigures" or disables the functionality associated with that statement or identifier.

If you do not specify *statement-path* or *identifier*, the entire hierarchy starting at the current hierarchy level is removed.

Options *statement-path*—(Optional) Path to an existing statement or identifier. Include this if the statement or identifier to be deleted is not at the current hierarchy level.

identifier—(Optional) Name of the statement or identifier to delete.

Usage Guidelines See "Removing a Statement from the Configuration" on page 82.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

See Also deactivate on page 197

edit

Syntax edit *statement-path*

Release Information Command introduced before JUNOS Release 7.4.

Description Move inside the specified statement hierarchy. If the statement does not exist, it is created.

You cannot use the edit command to change the value of identifiers. You must use the set command.

Options *statement-path*—Path to the statement.

Usage Guidelines See "Displaying the Current Configuration" on page 79.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

See Also set on page 205

exit

Syntax exit <configuration-mode>

Release Information Command introduced before JUNOS Release 7.4.

Description Exit the current level of the statement hierarchy, returning to the level prior to the last edit command, or exit from configuration mode. The quit and exit commands are synonyms.

Options none—Return to the previous edit level. If you are at the top of the statement hierarchy, exit configuration mode.

configuration-mode—(Optional) Exit from configuration mode.

Usage Guidelines See "Displaying the Current Configuration" on page 79.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

See Also top on page 209, up on page 209

help

Syntax help < (apropos | topic | reference) <*string*> >

Release Information Command introduced before JUNOS Release 7.4.

Description Display help about available configuration statements.

Options apropos—Display all hierarchy levels containing the statement.

reference—Display summary information for the statement.

string—String or regular expression matching configuration statements for which you need help.

topic—(Optional) Display usage guidelines for the statement.

Entering the help command without an option provides introductory information on how to use the help command.

Usage Guidelines See "Getting Help Based on a String in a Statement Name" on page 35.

Required Privilege Level configure—To enter configuration mode.

insert

Syntax insert <*statement-path*> *identifier1* (before | after) *identifier2*

Release Information Command introduced before JUNOS Release 7.4.

Description Insert an identifier in to an existing hierarchy.

Options after—Place *identifier1* after *identifier2*.

before—Place *identifier1* before *identifier2*.

identifier1—Existing identifier.

identifier2—New identifier to insert.

statement-path—(Optional) Path to the existing identifier.

Usage Guidelines See "Inserting a New Identifier" on page 86.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

load

Syntax load (patch | merge | override | replace | set | update) (*filename* | terminal) <relative>

Release Information Command introduced before JUNOS Release 7.4.

Description Load a configuration from an ASCII configuration file or from terminal input. Your current location in the configuration hierarchy is ignored when the load operation occurs.

Options *filename*—Name of the file to load. For information about specifying the filename, see "Specifying Filenames and URLs" on page 60.

merge—Combine the configuration that is currently shown in the CLI and the configuration in *filename*.

override—Discard the entire configuration that is currently shown in the CLI and load the entire configuration in *filename*. Marks every object as changed.

patch—Change part of the configuration and mark only those parts as changed.

replace—Look for a replace: tag in *filename*, delete the existing statement of the same name, and replace it with the configuration in *filename*.

set—Merge a set of commands with an existing configuration. This option executes the configuration instructions line-by-line as they are stored in a file or from a terminal. The instructions can contain any configuration mode command, such as set, edit, exit, and top.

relative—(Optional) Use the merge or replace option without specifying the full hierarchy level.

terminal—Use the text you type at the terminal as input to the configuration. Type Ctrl+d to end terminal input.

update—Discard the entire configuration that is currently shown in the CLI, and load the entire configuration in *filename.* Marks changed objects only.

Usage Guidelines See "Loading a Configuration From a File" on page 117.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

quit

Syntax quit <configuration-mode>

Release Information Command introduced before JUNOS Release 7.4.

Description Exit the current level of the statement hierarchy, returning to the level prior to the last edit command, or exit from configuration mode. The quit and exit commands are synonyms.

Options none—Return to the previous edit level. If you are at the top of the statement hierarchy, exit configuration mode.

configuration-mode—(Optional) Exit from configuration mode.

Usage Guidelines See "Displaying the Current Configuration" on page 79.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

See Also top on page 209, up on page 209

rename

Syntax rename <*statement-path*> *identifier1* to *identifier2*

Release Information Command introduced before JUNOS Release 7.4.

Description Rename an existing configuration statement or identifier.

Options *identifier1*—Existing identifier to rename.

identifier2—New name of identifier.

statement-path—(Optional) Path to an existing statement or identifier.

Usage Guidelines See "Renaming an Identifier" on page 85.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

replace

Syntax replace pattern *pattern1* with *pattern2* <upto *n*>

Release Information Command introduced in JUNOS Release 7.6.

Description Replace identifiers or values in a configuration.

Options *pattern1*—Text string or regular expression that defines the identifiers or values you want to match.

pattern2—Text string or regular expression that replaces the identifiers and values located with *pattern1*.

Juniper Networks uses standard UNIX-style regular expression syntax (as defined in POSIX 1003.2). If the regular expression contains spaces, operators, or wildcard characters, enclose the expression in quotation marks. Greedy qualifiers (match as much as possible) are supported. Lazy qualifiers (match as little as possible) are not.

upto *n*—Number of objects replaced. The value of *n* controls the total number of objects that are replaced in the configuration (not the total number of times the pattern occurs). Objects at the same hierarchy level (siblings) are replaced first. Multiple occurrences of a pattern within a given object are considered a single replacement. If you do not specify an upto option, all identifiers and values in the configuration that match *pattern1* are replaced.

Usage Guidelines See “Using Global Replace in a Configuration” on page 145.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

rollback

Syntax rollback (number | rescue)

Release Information Command introduced before JUNOS Release 7.4.

Description Return to a previously committed configuration. The software saves the last 50 committed configurations, including the rollback number, date, time, and name of the user who issued the commit configuration command.

The currently operational JUNOS software configuration is stored in the file juniper.conf, and the last three committed configurations are stored in the files juniper.conf.1, juniper.conf.2, and juniper.conf.3. These four files are located in the directory /config, which is on the router's flash drive. The remaining 46 previous committed configurations, the files juniper.conf.4 through juniper.conf.49, are stored in the directory /var/db/config, which is on the router's hard disk.

Options none—Return to the most recently saved configuration.

number—Configuration to return to.
Range: 0 through 49. The most recently saved configuration is number 0, and the oldest saved configuration is number 49.
Default: 0

rescue—Return to the rescue configuration.

Usage Guidelines See "Returning to a Configuration Prior to the Most Recently Committed One" on page 111 and "Creating and Returning to a Rescue Configuration" on page 114

Required Privilege Level rollback—To roll back to configurations other than the one most recently committed.

run

Syntax run *command*

Release Information Command introduced before JUNOS Release 7.4.

Description Run a top-level CLI command without exiting from configuration mode.

Options *command*—CLI top-level command.

Usage Guidelines See "Verifying a Configuration" on page 92.

Required Privilege Level configure—To enter configuration mode.

save

Syntax save *filename*

Release Information Command introduced before JUNOS Release 7.4.

Description Save the configuration to an ASCII file. The contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it. This allows a section of the configuration to be saved, while fully specifying the statement hierarchy.

When saving a file to a remote system, the software uses the scp/ssh protocol.

Options *filename*—Name of the saved file. You can specify a filename in one of the following ways:

- *filename*—File in the user's home directory (the current directory) on the local flash drive.
- *path/filename*—File on the local flash drive.
- /var/*filename* or /var/*path/filename*—File on the local hard disk.
- a:*filename* or a:*path/filename*—File on the local drive. The default path is / (the root-level directory). The removable media can be in MS-DOS or UNIX (UFS) format.
- *hostname:/path/filename, hostname:filename, hostname:path/filename,* or scp://*hostname/path/filename*—File on an scp/ssh client. This form is not available in the worldwide version of the JUNOS software. The default path is the user's home directory on the remote system. You can also specify *hostname* as *username@hostname*.
- ftp://*hostname*/*path*/*filename*—File on an FTP server. You can also specify *hostname* as *username@hostname* or *username*:*password@hostname*. The default path is the user's home directory. To specify an absolute path, the path must start with the string %2F; for example, ftp://*hostname*/%2F*path/filename*. To have the system prompt you for the password, specify prompt in place of the password. If a password is required, and you do not specify the password or prompt, an error message is displayed:

  user@host> **file copy ftp://***username***@ftp.***hostname***.net//***filename*
  file copy ftp.*hostname*.net: Not logged in.

  user@host> **file copy ftp://***username***:prompt@ftp.***hostname***.net//***filename*
  Password for *username*@ftp.*hostname*.net:

- http://*hostname*/*path*/*filename*—File on a Hypertext Transfer Protocol (HTTP) server. You can also specify *hostname* as *username@hostname* or *username*:*password@hostname*. If a password is required and you omit it, you are prompted for it.
- re0:*/path/filename* or re1:*/path/filename*—File on a local Routing Engine.

Usage Guidelines See "Deactivating and Reactivating Statements and Identifiers" on page 88.

Required Privilege Level configure—To enter configuration mode.

set

Syntax set <*statement-path*> *identifier*

Release Information Command introduced before JUNOS Release 7.4.

Description Create a statement hierarchy and set identifier values. This is similar to edit except that your current level in the hierarchy does not change.

Options *identifier*—Name of the statement or identifier to set.

statement-path—(Optional) Path to an existing statement hierarchy level. If that hierarchy level does not exist, it is created.

Usage Guidelines See "Displaying the Current Configuration" on page 79.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

See Also edit on page 198

show

Syntax show <*statement-path*> <*identifier*>

Release Information Command introduced before JUNOS Release 7.4.

Description Display the current configuration.

Options none—Display the entire configuration at the current hierarchy level.

identifier—(Optional) Display the configuration for the specified identifier.

statement-path—(Optional) Display the configuration for the specified statement hierarchy path.

Usage Guidelines See "Displaying the Current Configuration" on page 79.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

See the following sections:

show | display inheritance defaults

Syntax show | display inheritance defaults <| grep compress>

Release Information Command introduced before JUNOS Release 7.4.

Description Display the JUNOS software defaults that have been applied to the configuration.

Usage Guidelines See "Using JUNOS Default Groups" on page 176.

Options | grep compress—Display information about the compression of the current operational configuration.

Required Privilege Level view

Sample Output
```
user@host# show system ports | display inheritance defaults
## ## 'console' was inherited from group 'junos-defaults'
## 'vt100' was inherited from group 'junos-defaults'
## console type vt100;
```

show | display set

Syntax show | display set

Release Information Command introduced before JUNOS Release 7.4.

Description Display the configuration as a series of configuration mode commands required to recreate the configuration from the top level of the hierarchy as set commands

Usage Guidelines See "Displaying set Commands from the Configuration" on page 103.

Required Privilege Level view

Sample Output
```
user@host# show | display set
set interfaces fe-0/0/0 unit 0 family inet address 192.168.1.230/24
set interfaces fe-0/0/0 unit 0 family iso
set interfaces fe-0/0/0 unit 0 family mpls
set interfaces fe-0/0/0 unit 1 family inet address 10.0.0.1/8
deactivate interfaces fe-0/0/0 unit 1
```

show | display set relative

Syntax show | display set relative

Release Information Command introduced before JUNOS Release 7.4.

Description Display the configuration as a series of configuration mode commands required to recreate the configuration from the current hierarchy level.

Usage Guidelines See "Displaying set Commands from the Configuration" on page 103.

Required Privilege Level view

Sample Output

```
[edit interfaces fe-0/0/0]
user@host# show
unit 0 {
    family inet {
        address 192.107.1.230/24;
    }
    family iso;
    family mpls;
}
inactive: unit 1 {
    family inet {
        address 10.0.0.1/8;
    }
}
user@host# show | display set relative
set unit 0 family inet address 192.107.1.230/24
set unit 0 family iso
set unit 0 family mpls
set unit 1 family inet address 10.0.0.1/8
deactivate unit 1
```

show groups junos-defaults

Syntax show groups junos-defaults

Release Information Command introduced before JUNOS Release 7.4.

Description Display the full set of available preset statements from the JUNOS software default group.

Usage Guidelines See "Using JUNOS Default Groups" on page 176.

Required Privilege Level view

Sample Output

```
user@host# show groups junos-defaults
groups {
    junos-defaults {
        applications {
            #
            # File Transfer Protocol
            #
            application junos-ftp {
                application-protocol ftp;
                protocol tcp;
                destination-port 21;
            }
            #
            # Trivial File Transfer Protocol
            #
            application junos-tftp {
                application-protocol tftp;
                protocol udp;
                destination-port 69;
            }
            #
            # RPC port mapper on TCP
            #
            application junos-rpc-portmap-tcp {
                application-protocol rpc-portmap;
                protocol tcp;
                destination-port 111;
            }
            #
            # RPC port mapper on UDP
            #
        }
    }
}
```

status

Syntax status

Release Information Command introduced before JUNOS Release 7.4.

Description Display the users currently editing the configuration.

Usage Guidelines See “Displaying Users Currently Editing the Configuration” on page 99.

Required Privilege Level configure—To enter configuration mode.

top

Syntax top <*configuration-command*>

Release Information Command introduced before JUNOS Release 7.4.

Description Return to the top level of configuration command mode, which is indicated by the [edit] banner.

Option *configuration-command*—Issue configuration mode commands from the top of the hierarchy.

Usage Guidelines See “Displaying the Current Configuration” on page 79 and “Displaying the Current Configuration” on page 79.

Required Privilege Level configure—To enter configuration mode.

See Also exit on page 198, up on page 209

up

Syntax up <*number*> <*configuration-command*>

Release Information Command introduced before JUNOS Release 7.4.

Description Move up one level in the statement hierarchy.

Options none—Move up one level in the configuration hierarchy.

number—(Optional) Move up the specified number of levels in the configuration hierarchy.

configuration-command—Issue configuration mode commands from a location higher in the hierarchy.

Usage Guidelines See “Displaying the Current Configuration” on page 79 and “Displaying the Current Configuration” on page 79.

Required Privilege Level configure—To enter configuration mode.

See Also exit on page 198, top on page 209

update

Syntax update

Release Information Command introduced in JUNOS Release 7.5.

Description Update private candidate configuration with a copy of the most recently committed configuration, including your private changes.

Usage Guidelines See "Updating the Configure Private Configuration" on page 103.

NOTE: The update command is available only when you are in configure private mode.

wildcard

Syntax wildcard delete <*statement-path*> <*identifier*> <*regular-expression*>

Release Information Command introduced before JUNOS Release 7.4.

Description Delete a statement or identifier. All subordinate statements and identifiers contained within the specified statement path are deleted with it.

Deleting a statement or an identifier effectively "unconfigures" or disables the functionality associated with that statement or identifier.

If you do not specify *statement-path* or *identifier*, the entire hierarchy starting at the current hierarchy level is removed.

Options delete—Delete several related configuration items simultaneously, such as channelized interfaces or static routes, by using a single command and regular expressions.

statement-path—(Optional) Path to an existing statement or identifier. Include this if the statement or identifier to be deleted is not at the current hierarchy level.

identifier—(Optional) Name of the statement or identifier to delete.

Usage Guidelines See "Use the opto option to perform a replacement:" on page 149.

Required Privilege Level configure—To enter configuration mode; other required privilege levels depend on where the statement is located in the configuration hierarchy.

Chapter 15

Summary of CLI Operational Mode Commands

The following sections explain each of the command-line interface (CLI) operational mode commands described in this book. The commands are organized alphabetically.

configure

Syntax configure | configure exclusive | configure private

Release Information Command introduced before JUNOS Release 7.4.

Description Enter configuration mode.

Usage Guidelines See "Entering and Exiting Configuration Mode" on page 76.

Required Privilege Level configure

file

Syntax file (archive | checksum | compare | copy | delete | list | rename | show)

Release Information Command introduced before JUNOS Release 7.4.

Description Copy files to and from the router, compare files, or delete a file on a local router.

Usage Guidelines See "Viewing Files and Directories" on page 57. See also the *JUNOS System Basics and Services Command Reference*.

Required Privilege Level maintenance

help

Syntax help <(topic | reference | syslog | tip (cli <*number*>))>

Release Information Command introduced before JUNOS Release 7.4.

Description Display help about available configuration statements or general information on getting help.

Options Display all hierarchy levels containing the statement.

reference—Display summary information for the statement.

syslog—Display system log messages.

tip cli <*number*> —Display CLI tips that are associated with a number.

topic—(Optional) Display usage guidelines for the statement.

Entering the help command without an option provides introductory information on using the help and ? commands.

Usage Guidelines See "Getting Help Based on a String in a Statement Name" on page 35 and "Displaying Tips About CLI Commands" on page 41.

Required Privilege Level None.

| (pipe)

Syntax | (compare | count | display (changed | commit-scripts | detail | display set | inheritance | xml) | except *pattern* | find *pattern* | hold | last *lines* | match *pattern* | no-more | request message (all | *account@terminal*) resolve <full-names> | save *filename* | trim *columns*)

Release Information Command introduced before JUNOS Release 7.4.
display commit-scripts option added in JUNOS Release 7.4.

Description Filter the output of an operational mode or a configuration mode command.

Options compare (filename | rollback *n*)—(Configuration mode only, and only with the show command) Compare configuration changes with another configuration file.

count—Display the number of lines in the output.

display—Display additional information about the configuration contents.

- changed—Tag changes with junos:changed attribute (XML only).
- commit-scripts—(Configuration mode only) Display all statements that are in a configuration, including statements that were generated by transient changes. For more information, see the *JUNOS Configuration and Diagnostic Automation Guide*.
- detail—(Configuration mode only) Display configuration data detail.

- inheritance <brief | default | groups | terse>—(Configuration mode only) Display inherited configuration data and source group.
- set—Display the configuration as a series of configuration mode commands required to recreate the configuration.
- xml—(Operational mode only) Display the command output as JUNOScript (Extensible Markup Language [XML]) tags.

except *pattern*—Ignore text matching a regular expression when searching the output. If the regular expression contains spaces, operators, or wildcard characters, enclose it in quotation marks.

find *pattern*—Display the output starting at the first occurrence of text matching a regular expression. If the regular expression contains spaces, operators, or wildcard characters, enclose it in quotation marks (" ").

last *lines*—Display the last number of lines you want to view from the end of the configuration.

hold—Hold text without exiting the –More– prompt.

match *pattern*—Search for text matching a regular expression. If the regular expression contains spaces, operators, or wildcard characters, enclose it in quotation marks.

no-more—Display output all at once rather than one screen at a time.

resolve—Convert IP addresses into Domain Name System (DNS) names. Truncates to fit original size unless full-names is specified. To prevent the names from being truncated, use the full-names option.

request message (all | *account@terminal*)—Display command output on the terminal of a specific user logged in to your router, or on the terminals of all users logged in to your router.

save *filename*—Save the output to a file or URL. For information about specifying the filename, see "Specifying Filenames and URLs" on page 60.

trim *columns*—Trim specified number of columns from the start line.

Usage Guidelines See "Filtering Command Output" on page 127 and "Displaying the Current Configuration" on page 79.

quit

Syntax quit

Release Information Command introduced before JUNOS Release 7.4.

Description Exit from the CLI to a UNIX shell.

Required Privilege Level shell and maintenance

request

Syntax request <chassis | ipsec switch | message | mpls | routing-engine | security | services | system | flow-collector | support information>

Release Information Command introduced before JUNOS Release 7.4.

Description Stop or reboot router components, switch between primary and backup components, display messages, and display system information.

Usage Guidelines Most request commands are discussed in the *JUNOS System Basics and Services Command Reference*. The following request commands are discussed in the *JUNOS Interfaces Command Reference*: request ipsec switch and request services.

Required Privilege Level maintenance

restart

Syntax restart
<adaptive-services | chassis-control | class-of-service | disk-monitoring | ecc-error-logging | firewall | interface-control | kernel-replication | l2tpd-service | mib-process | network-access-service | pgm | pic-services-logging | pppoe | remote-operations | routing <logical-router *logical-router-name*> | sampling | service-deployment | snmp | web-management>
<gracefully | immediately | soft>

Release Information Command introduced before JUNOS Release 7.4.

Description Restart router software processes on all platforms (with the exception of routing matrixes and J-series Services Routers).

Usage Guidelines See the *JUNOS System Basics and Services Command Reference*.

Required Privilege Level reset

restart (Routing Matrix)

Syntax restart
<adaptive-services | chassis-control | class-of-service | disk-monitoring | ecc-error-logging | firewall | interface-control | kernel-replication | l2tpd-service | mib-process | network-access-service | pgm | pic-services-logging | pppoe | remote-operations | routing <logical-router *logical-router-name*> | sampling | service-deployment | snmp | web-management>
<all | all-lcc | lcc *number*>
<gracefully | immediately | soft>

Release Information Command introduced before JUNOS Release 7.4.

Description Restart router software processes on a routing matrix.

Usage Guidelines See the *JUNOS System Basics and Services Command Reference.*

Required Privilege Level reset

restart (J-series Services Routers)

Syntax restart
<adaptive-services | chassis-control | class-of-service | dhcp | firewall | interface-control | l2tpd-service | mib-process | network-access-service | pgm | pppoe | remote-operations | routing <logical-router *logical-router-name*> | sampling | service-deployment | snmp | usb-control | web-management>
<gracefully | immediately | soft>

Release Information Command introduced before JUNOS Release 7.4.

Description Restart router software processes on J-series Services Routers.

Usage Guidelines See the *JUNOS System Basics and Services Command Reference.*

Required Privilege Level reset

set

Syntax set (chassis | cli | date)

Release Information Command introduced before JUNOS Release 7.4.

Description Configure chassis, CLI properties, and the router's date and time.

Usage Guidelines See "Controlling the CLI Environment" on page 137 and "Using the Comment Character #" on page 68. For information about setting chassis properties, see the *JUNOS System Basics and Services Command Reference.*

Required Privilege Level view

show

Syntax show (accounting | aps | arp | as-path | bfd | bgp | chassis | class-of-service | cli | configuration | connections | dvmrp | dynamic-tunnels | firewall | helper | host | igmp | ike | ilmi | interfaces | ipsec | ipv6 | isis | l2circuit | l2vpn | ldp | link-management | log | mld | mpls | msdp | multicast | ntp | ospf | ospf3 | passive-monitoring | pfe | pgm | pim | policer | policy | pppoe | rip | ripng | route | rsvp | sap | services | snmp | ssh-known-hosts | system | task | ted | version | vpls | vrrp)

Release Information Command introduced before JUNOS Release 7.4.

Description Show information about all aspects of the software, including interfaces and routing protocols.

Usage Guidelines Most show commands are discussed in the *JUNOS System Basics and Services Command Reference*. The following show commands are discussed in the *JUNOS Interfaces Command Reference*: show aps, show ike, show ilmi, show interfaces, show ipsec, show passive-monitoring, show pppoe, show services, and show vrrp.

Required Privilege Level Depends on the specific command.

Part 5
Indexes

Index

Symbols

A

C

D

E

F

G

H

R

S

T

U

W

X

Index of Statements and Commands

T

U

W