SRC-PE Software

CLI User Guide

Release 2.0.x

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

Part Number: 530-022134-01, Revision 01

Products made or sold by Juniper Networks (including the ERX-310, ERX-705, ERX-710, ERX-1410, ERX-1440, M5, M7i, M10, M10i, M20, M40, M40e, M160, M320, and T320 routers, T640 routing node, and the JUNOS, JUNOSe, and SDX-300 software) or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

SRC-PE Software CLI User Guide, Release 2.0.x
Writing: Linda Creed, Lisa Kelly
Editing: Fran Mues, Stella Hackell
Illustration: Nathaniel Woodward
Cover Design: Edmonds Design

Revision History
11 October 2007—Revision 1

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

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)).

Table of Contents

Part 2 **Operational Mode and Configuration Mode**

Part 3 SRC CLI Environment and SRC Component Management

About This Guide

This preface provides the following guidelines for using the *SRC-PE Software CLI User Guide.*

Objectives

This guide provides information about the Session and Resource Control (SRC) command-line interface (CLI). It describes features of the CLI, explains how to get started using the CLI, and describes basic CLI tasks.

NOTE: If the information in the latest *SRC Release Notes* differs from the information in this guide, follow the *SRC Release Notes*.

Audience

This guide is intended for experienced system and network specialists working with JUNOSe routers and JUNOS routing platforms in an Internet access environment. We assume that readers know how to use the routing platforms, directories, and RADIUS servers that they will deploy in their SRC networks. For users who deploy the SRC software on a Solaris platform, we also assume that readers are familiar with the Lightweight Directory Access Protocol (LDAP) and the UNIX operating system.

If you are using the SRC software in a cable network environment, we assume that you are familiar with the *PacketCable Multimedia Specification* (PCMM) as defined by Cable Television Laboratories, Inc. (CableLabs) and with the Data-over-Cable Service Interface Specifications (DOCSIS) 1.1 protocol. We also assume that you are familiar with operating a multiple service operator (MSO) multimedia-managed IP network.

Documentation Conventions

The sample screens used throughout this guide are representations of the screens that are displayed when you install and configure the SRC software. The actual screens may differ.

For convenience and clarity, the installation and configuration examples show default file paths. If you do not accept the installation defaults, your paths will vary from the examples.

Table 1 defines notice icons used in this guide. Table 2 defines text conventions used throughout the documentation.

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.
⚡	Warning	Alerts you to the risk of personal injury.

Table 2: Text Conventions

Convention	Description	Examples
Bold typeface	■ Represents keywords, scripts, and tools in text. ■ Represents a GUI element that the user selects, clicks, checks, or clears.	■ Specify the keyword **exp-msg**. ■ Run the **install.sh** script. ■ Use the **pkgadd** tool. ■ To cancel the configuration, click **Cancel**.
Bold sans serif typeface	Represents text that the user must type.	user@host# **set cache-entry-age** *cache-entry-age*
`Monospace sans serif typeface`	Represents information as displayed on your terminal's screen, such as CLI commands in output displays.	

```
nic-locators {
  login {
    resolution {
      resolver-name /realms/login/A1;
      key-type LoginName;
      value-type SaeId;
    }
```

Table 2: Text Conventions (continued)

Convention	Description	Examples
Regular sans serif typeface	■ Represents configuration statements. ■ Indicates SRC CLI commands and options in text. ■ Represents examples in procedures. ■ Represents URLs.	■ system ldap server { stand-alone; ■ Use the request sae modify device failover command with the force option. ■ user@host# . . . ■ http://www.juniper.net/techpubs/software/management/sdx/api-index.html
Italic sans serif typeface	Represents variables in SRC CLI commands.	user@host# **set local-address** *local-address*
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <gfwif>.
Key name	Indicates the name of a key on the keyboard.	Press Enter.
Key names linked with a plus sign (+).	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
Italic typeface	■ Emphasizes words. ■ Identifies chapter, appendix, and book names. ■ Identifies distinguished names. ■ Identifies files, directories, and paths in text but not in command examples.	■ There are two levels of access: *user* and *privileged.* ■ *Chapter 2, Services.* ■ *o = Users, o = UMC* ■ The */etc/default.properties* file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	Plugin.radiusAcct-1.class = \ net.juniper.smgt.sae.plugin\ RadiusTrackingPluginEvent
Words separated by the | symbol	Represent a choice to select one keyword or variable to the left or right of this symbol. (The keyword or variable may be either optional or required.)	diagnostic | line

Related Juniper Networks Documentation

With each SRC software release, we provide the *SRC Documentation CD,* which contains the documentation described in Table 3.

With each SRC Application Library release, we provide the *SRC Application Library CD*. This CD contains both the software applications and the *SRC Application Library Guide.*

A complete list of abbreviations used in this document set, along with their spelled-out terms, is provided in the *SRC-PE Getting Started Guide.*

Table 3: Juniper Networks C-series and SRC Technical Publications

Document	Description
Core Documentation Set	
C2000 and C4000 Hardware Guide	Describes the hardware platforms and how to install, maintain, replace, and troubleshoot them. The guide also includes specifications.
C2000 and C4000 Quick Start Guide	Describes how to get the C-series Controller up and running quickly. Intended for experienced installers who want to expedite the installation process.
SRC-PE Getting Started Guide	Describes the SRC software, how to set up an initial software configuration, and how to upgrade the SRC software. It also explains how to manage a C-series Controller. The guide describes how to set up and start the SRC CLI and the C-Web interface, as well as other SRC configuration tools. It includes reference material for the SRC documentation.
SRC-PE CLI User Guide	Describes how to use the SRC CLI, configure and monitor the platform with the CLI, and control the CLI environment. The guide also describes how to manage SRC components with the CLI.
SRC-PE Network Guide: SAE, Juniper Networks Routers, NIC, and SRC-ACP	Describes how to use and configure the SAE, the NIC, and the SRC-ACP (Admission Control Plug-In) application. This guide also provides detailed information for using JUNOSe routers and JUNOS routing platforms in the SRC network.
SRC-PE Integration Guide: Network Devices, Directories, and RADIUS Servers	Describes how to integrate external components—network devices, directories, and RADIUS servers—into the SRC network. The guide provides detailed information about integrating specific models of the external components.
SRC-PE Services and Policies Guide	Describes how to work with services and policies. The guide provides an overview, configuration procedures, and management information. The guide also provides information about the SRC tools for configuring policies.
SRC-PE Subscribers and Subscriptions Guide	Describes how to work with residential and enterprise subscribers and subscriptions. The guide provides an overview, configuration procedures, and management information. This guide also provides information about the enterprise service portals, including the Enterprise Manager Portal.
SRC-PE Monitoring and Troubleshooting Guide	Describes how to use logging, the SNMP agent, the SRC CLI, and the C-Web interface to monitor and troubleshoot SRC components. This guide also describes the SNMP traps.
SRC-PE Solutions Guide	Provides high-level instructions for SRC implementations. The guide documents the following scenarios: managing QoS services on JUNOSe routers; managing subscribers in a wireless roaming environment; providing voice over IP (VoIP) services; integrating the SRC software in a PCMM environment, including the use of the Juniper Policy Server (JPS); and mirroring subscriber traffic on JUNOSe routers.
SRC-PE CLI Command Reference, Volume 1 *SRC-PE CLI Command Reference, Volume 2*	Together constitute information about command and statement syntax; descriptions of commands, configuration statements, and options; editing level of statement options; and a history of when a command was added to the documentation.
SRC-PE NETCONF API Guide	Describes how to use the NETCONF application programming interface (API) to configure or request information from the NETCONF server on a C-series Controller that runs the SRC software.
SRC-PE XML API Configuration Reference	Describes the tag elements in the SRC Extensible Markup Language (XML) application programming interface (API) that are equivalent to configuration statements in the SRC command-line interface (SRC CLI).
SRC-PE XML API Operational Reference	Describes the tag elements in the SRC Extensible Markup Language (XML) application programming interface (API) that are equivalent to operational commands in the SRC command-line interface (SRC CLI).

Table 3: Juniper Networks C-series and SRC Technical Publications (continued)

Document	Description
SRC-PE Comprehensive Index	Provides a complete index of the SRC guides, excluding the *C-series Hardware Guide*, the *SRC CLI Command Reference,* the *SRC-PE NETCONF API Guide,* the *SRC-PE XML API Configuration Reference,* and the *SRC-PE XML API Operational Reference.*
Application Library	
SRC Application Library Guide	Describes how to install and work with applications that you can use to extend the capabilities of the SRC software. The guide documents the following applications: SRC-SG (SOAP Gateway) Web applications, an application to provide threat mitigation, an application to provide tracking and QoS control at the application level by integrating the SRC software with the Ellacoya deep packet inspection (DPI) platform, and an application to control volume usage.
Release Notes	
SRC-PE Release Notes *SRC Application Library Release Notes*	In the *Release Notes*, you will find the latest information about features, changes, known problems, resolved problems, supported platforms and network devices (such as Juniper Networks routers and CMTS devices), and third-party software. If the information in the *Release Notes* differs from the information found in the documentation set, follow the *Release Notes.* Release notes are included in the corresponding software distribution and are available on the Web.

Obtaining Documentation

To obtain the most current version of all Juniper Networks technical documentation, see the products documentation page on the Juniper Networks Web site at

http://www.juniper.net/

To order printed copies of this manual and other Juniper Networks technical documents or to order a documentation CD, which contains this manual, contact your sales representative.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation to better meet your needs. 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 SRC CLI

The SRC command-line interface (CLI) is the software interface you use to configure the C-series Controller. You can also use the CLI to configure supported components for SRC software installed on Solaris platforms. This chapter provides an overview of the SRC CLI. Topics include:

Overview of the SRC CLI

The SRC CLI is a JUNOS-like command shell that runs on top of a Linux-based operating system kernel on a C-series Controller or on top of a Solaris operating system. By leveraging industry-standard tools and utilities, the CLI provides a powerful set of commands you can use to monitor and configure the SRC software and a C-series Controller.

The SRC 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 supports Emacs-style keyboard sequences that allow you to move around on a command line and scroll through recently executed commands.

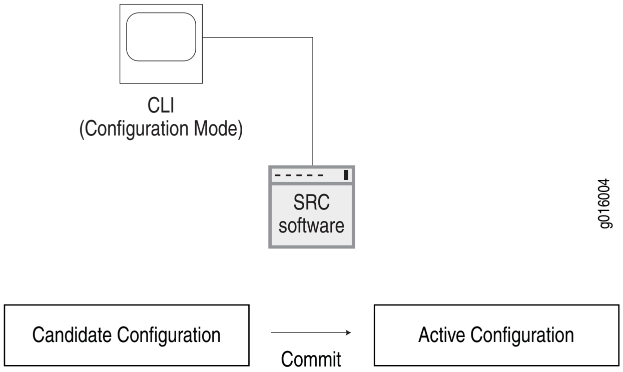
Understanding SRC CLI Command Modes

The SRC CLI has two modes:

- Operational mode—Provides commands to monitor and troubleshoot system status.
- Configuration mode—Provides commands and configuration statements to define properties for the SRC software. The statements appear in a hierarchy which groups related properties.

In configuration mode, you view and change a working configuration, called the *candidate configuration*. This configuration allows you to make configuration changes without causing operational changes to the current operating configuration, called the *active configuration*. The software does not implement the changes in a candidate configuration until you commit them, which activates the configuration in the SRC software.

Figure 1: Committing a Configuration



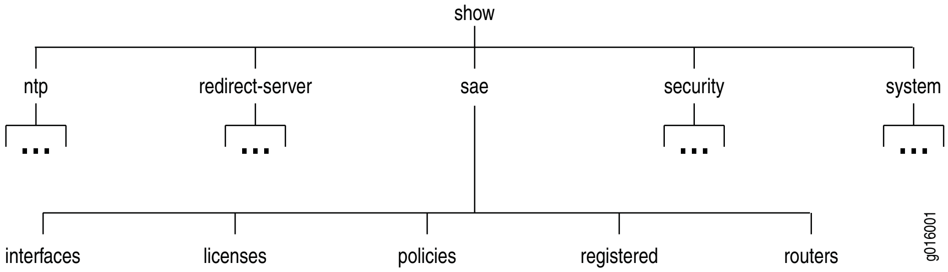
Understanding SRC Command and Statement Hierarchies

The SRC CLI provides numerous commands and statements and organizes them in a hierarchical fashion.

SRC 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. Figure 2 illustrates a portion of the show command hierarchy.

Figure 2: CLI Command Hierarchy Example



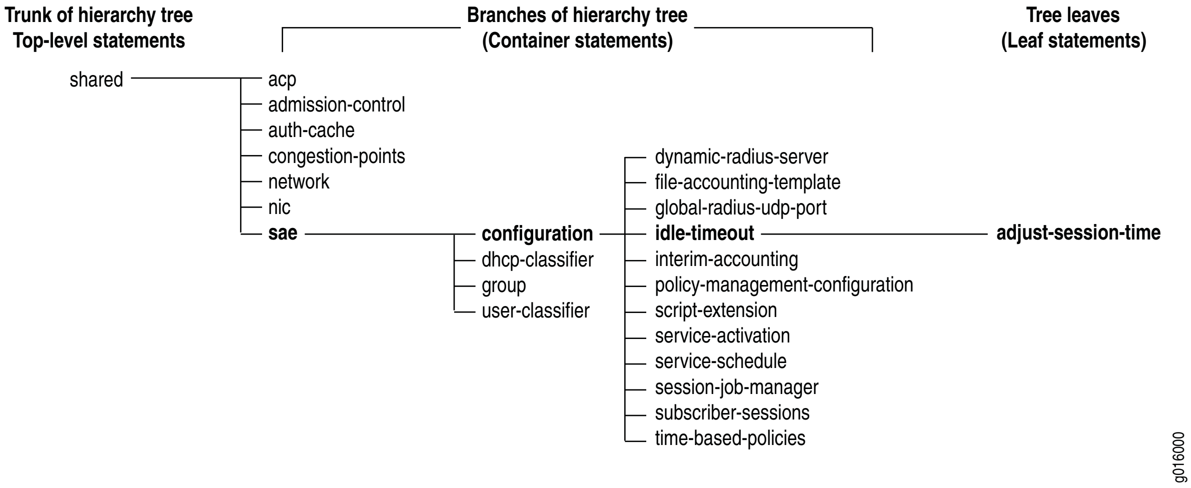
To execute a command, you enter the full command name, starting at the top level of the hierarchy. For example, to display information about SAE licenses, you enter the show sae licenses command.

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 the container and leaf statements together form the *configuration hierarchy*.

Figure 3 illustrates part of the hierarchy tree.

Figure 3: Configuration Statement Hierarchy Example



Key Features of the SRC CLI

The hierarchical organization results in commands that have a regular syntax and provides several features that simplify SRC 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 SRC software, JUNOS software, or routing software, you can use many of the CLI commands without referring to the documentation.
- Detailed descriptions of command and configuration statements—Complete information about commands and statements from the **help** command. You can access the reference documentation for each command and statement by typing the **help** command followed by the command or **help configuration** followed by the configuration statement.
- 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 Spacebar or the tab key. If the partially typed letters begin a string that uniquely identifies a command, the complete command name appears. Otherwise, a caret (^) indicates that you have entered an ambiguous command, and the possible completions are displayed. Completion also applies to other strings, such as filenames, interface names, usernames, and configuration statements.

Leveraging Industry-Standard Technologies

The operating system on a C-series Controller is based on a Linux kernel, with a special shell called the CLI (command-line interface). If you run the SRC software on a Solaris platform, the software leverages the features of a Solaris kernel. A variety of standard utilities are available. For example, you can:

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

Chapter 2

Getting Started: A Quick Tour of the SRC CLI

As an introduction to the SRC command-line interface (CLI), this chapter provides instructions for simple steps that you take after installing software on the system. It shows you how to start the CLI, view the command hierarchy, and make small configuration changes. Topics include:

Before You Start the SRC CLI

Make sure the SRC software has been configured for:

- C-series Controllers only—Remote access through SSH and/or Telnet

> **NOTE:** On a C-series Controller, `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.

- All platforms—A user account that has superuser privileges

Related Topics

For information about initial CLI configuration, see:

- C-series Controller—*C-series Hardware Guide* and *SRC Getting Started Guide*
- SRC software on a Solaris platform—*SRC Getting Started Guide*

Starting the SRC CLI

When you log in to the CLI, the privileges for your user account determine which commands and configuration statements you can access. A login account with superuser privileges gives a user access to all commands and statements.

Starting the SRC CLI on a C-series Controller

To log in to a C-series Controller and start the CLI through an admin account:

```
#ssh admin@my_C-series-controller
Password:

— SRC CLI 7.0 build CLI.A.7.0.0.0171
(c) 2005-2006 Juniper Networks Inc.
user@host>
```

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

Starting the SRC CLI on a Solaris Platform

To start the CLI on a Solaris platform:

1. Log in to the Solaris system through a user account that has superuser privileges configured for the SRC software:
2. Start the CLI from the directory in which it is installed:

```
# /opt/UMC/cli/bin/cli
— SRC CLI 7.0 build CLI.A.7.0.0.0171
(c) 2005-2006 Juniper Networks Inc.
user@host>
```

Displaying Commands

The SRC CLI includes several ways to get help about commands.

To use the various Help commands:

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

```
user@host> ?
Possible completions:
  clear                Delete statement
  configure            Enter configuration mode
  disable              Stop an SRC component
  enable               Start an installed SRC component
  exit                 Exit a CLI session
  file                 Perform file operations
  help                 Display help about commands and statements
  history              Display command history
  ping                 Ping remote target
  request              Request service
  restart              Restart an SRC component
  set                     Properties for the CLI environment
  show                    Display information
  ssh                     Open SSH session to another host
  start                   Start shell
  telnet                  Telnet to another host
  test                    Test a NIC resolution
  traceroute              Trace route to remote host
```

2. Type file ? to show all possible completions for the file command.

```
user@host> file ?
Possible completions:
  archive                 Archive files from the system (local)
  checksum                Calculate file checksum
  compare                 Compare files (local)
  copy                    Copy files
  delete                  Delete a file (local)
  list                    List files (local)
  rename                  Rename a file (local)
  show                    Show file contents
```

3. Type file archive ? to show all possible completions for the file archive command.

```
user@host> file archive ?
Possible completions:
  compress                Compresses the archived file using GNU gzip (.tgz)
* destination             Name of created archive
* source                  Path of directory to archive
```

For more information about getting help about commands and statements, see *Chapter 4, Getting Online Help for the SRC CLI*.

Verifying System Status

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

To help you become familiar with show commands:

1. Type show ? to display the list of show commands that you can use to monitor the SRC software and C-series Controller.

```
user@host> show ?
Possible completions:
  acp                     Display information about ACP
  cli                     Configure properties for the CLI environment
  component               Display information about SDX components
  configuration           Information about the SDX configuration
  date                    System time and date
  disk                    Configuration for RAID disks
  interfaces              Show interface information
  iptables                Display information about the iptables LINUX tool
  jps                     Display information about the JPS
  nic                     Display information about the NIC
  ntp                     NTP configuration information
  redirect-server         Statistics for redirect server
  route                   Show routing table information
  sae                     Display SAE information
  security            Display security information
  system              Display system information
```

2. Use the show component command to view a the status of installed components.

```
user@host> show component
Installed Components
Name      Version                                            Status
cli       Release: 7.0 Build: CLI.A.7.0.0.0171               running
acp       Release: 7.0 Build: ACP.A.7.0.0.0174               disabled
jdb       Release: 7.0 Build: DIRXA.A.7.0.0.0176             running
editor    Release: 7.0 Build: EDITOR.A.7.0.0.0176            disabled
redir     Release: 7.0 Build: REDIR.A.7.0.0.0176             disabled
licSvr    Release: 7.0 Build: LICSVR.A.7.0.0.0179            stopped
nic       Release: 7.0 Build: GATEWAY.A.7.0.0.0170           disabled
sae       Release: 7.0 Build: SAE.A.7.0.0.0166               running
www       Release: 7.0 Build: UMC.A.7.0.0.0169               disabled
jps       Release: 7.0 Build: JPS.A.7.0.0.0172               disabled
agent     Release: 7.0 Build: SYSMAN.A.7.0.0.0174            disabled
webadm    Release: 7.0 Build: WEBADM.A.7.0.0.0173            disabled
```

3. Use the show system information command to view general system information.

```
user@host> show system information

System Identification
Hostname         myC-seriesController
Manufacturer     Juniper Networks
Product Name     SDX-2000
Version          1.0
Serial Number    0207082006000001
UUID             48384441-5254-0030-4859-0030485977EE
Hostid           e30a2e07
Software version SDX-300 Release 7.0 [B.7.0.0-7]

System Time
Current time              2006-12-19 13:52:26 EST
Uptime                    21:30
Number of active users    3
Load Averages (1m/5m/15m) 0.02/0.07/0.02

Memory
Total 15G
Free  14G

CPU Info
Number of CPU 4
CPU Model     Dual Core AMD Opteron(tm) Processor 265
Clock Speed   1804.132 MHz

Disk Information
  Mountpoint       Total Used  Use%
  /                2015M  955M  47%
  /altroot         2015M   35M   1%
  /altvar            29G   75M   0%
  /boot              98M   14M  14%
  /var               31G  829M   2%
```

```
Temperature
System +24 C
CPU-1  +34 C
CPU-2  +36 C

Fan Speed
Fan-1 9375 RPM
Fan-2 9375 RPM
```

Configuring a User Account

This sample procedure describes how to use the CLI to view system status and to perform a simple configuration change. You configure a new user account, one for your own use or a test account.

To configure a user account on the system:

1. Log in as user with superuser privileges, start the CLI, and enter configuration mode.

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

   The prompt in brackets ([edit]), also known as a *banner*, shows that you are in configuration edit mode, at the top of the hierarchy.

2. Move to the [edit system login] level of the configuration hierarchy.

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

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

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

3. Add a new user account.

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

   This example adds an account nchen (for Nathan Chen), but you can use any account name.

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]
   user@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]
   user@host# set class super-user
   ```

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

   ```
   [edit system login user nchen]
   user@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, and then confirm the new password.

7. Commit the configuration.

   ```
   [edit system login user nchen]
   user@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, and then exit.

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

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

9. Log out of the SRC software.

   ```
   user@host> exit
   [user@host]#
   ```

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

    ```
    – SRC CLI 7.0 build CLI.A.7.0.0.0171
    (c) 2005-2006 Juniper Networks Inc.
    nchen@host>
    ```

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

Creating a Configuration

This sample procedure shows how to configure SAE properties for aggregate services as an example of how to navigate through various hierarchy levels in the CLI and use help and show commands to obtain information while working at the CLI.

In the SRC software, an aggregate service is a type of SAE service that comprises a number of individual services. Combining services lets the SRC software treat the services within an aggregate service as a unit.

The final configuration looks like this:

```
sae {
  configuration {
    aggregate-services {
      keepalive-time 172800;
      keepalive-retry-time 900;
      activation-deactivation-time 900;
      failed-notification-retry-time 92000;
    }
  }
}
```

To configure SAE properties for aggregate services:

1. Enter configuration mode.

   ```
   user@host> edit
   Entering configuration mode.

   [edit]
   user@host#
   ```

2. In configuration mode, move to the hierarchy level at which you configure aggregate services.

   ```
   [edit]
   user@host# edit shared sae configuration aggregate-services

   [edit shared sae configuration aggregate-services]
   user@host#
   ```

   Press the space bar after typing the initial characters of a word to quickly complete the word.

3. Verify which values you can set.

```
[edit shared sae configuration aggregate-services]
user@host# set ?
Possible completions:
  activation-deactivation-time
                       Time to activate or deactivate fragment service session (0..INF s)
  failed-notification-retry-time
                       Maximum time to send failure notifications (0..INF s)
  keepalive-retry-time Length of keepalive time period (0..INF s)
  keepalive-time       Keepalive from aggregate svce session to remote session (0..INF s)
```

4. Set the values for the four time intervals.

   ```
   [edit shared sae configuration aggregate-services]
   user@host# set keepalive-time 172800

   user@host# set keepalive-retry-time 900

   [edit]
   user@host# set activation-deactivation-time 900

   [edit]
   user@host# set failed-notification-retry-time 9200
   ```

5. Verify the configuration.

   ```
   [edit shared sae configuration aggregate-services]
   user@host# show
   keepalive-time 172800;
   keepalive-retry-time 900;
   activation-deactivation-time 900;
   failed-notification-retry-time 9200;
   ```

6. Move up one level in the hierarchy, and run the show command again.

   ```
   [edit shared sae configuration aggregate-services]
   user@host# up

   [edit shared sae configuration]
   user@host# show
   policy-management-configuration {
     enable-junose-classifier-expansion;
   }
   aggregate-services {
     keepalive-time 172800;
     keepalive-retry-time 900;
     activation-deactivation-time 900;
     failed-notification-retry-time 9200;
   }
   . . .
   ```

   The output shows the configuration for aggregate services plus any other configuration under the [shared sae configuration] hierarchy level.

7. Before you commit the configuration, verify that the configuration is correct.

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

8. Commit the configuration.

   ```
   [edit shared sae configuration]
   user@host# commit
   commit complete.
   ```

Using Shortcuts to Create a Configuration

You can navigate through the configuration hierarchy to enter statements, or you can use set commands to configure statements at the [edit] hierarchy level. The following list of commands creates the same configuration as the procedure:

```
user@host# set shared sae configuration aggregate-services keepalive-time 172800

user@host# set shared sae configuration aggregate-services
keepalive-retry-time 900

[edit]
user@host# set shared sae configuration aggregate-services
activation-deactivation-time 900

[edit]
user@host# set shared sae configuration aggregate-services
failed-notification-retry-time 92000
```

Making Changes to the Configuration

You can change a configuration by entering a set command and providing a new value.

To change the value for the keepalive timer for aggregate services:

1. Navigate to the location in the hierarchy where properties for aggregate services are configured.

   ```
   [edit]
   user@host#
   user@host# edit shared sae configuration aggregate-services

   [edit shared sae configuration aggregate-services]
   user@host#
   ```

2. Change the value for the keepalive timer.

   ```
   [edit shared sae configuration aggregate-services]
   user@host# set keepalive-time 150000
   ```

3. Verify the change.

   ```
   [edit shared sae configuration aggregate-services]
   user@host# show
   keepalive-time 150000;
   keepalive-retry-time 900;
   activation-deactivation-time 900;
   failed-notification-retry-time 9200;
   ```

4. Commit the change.

   ```
   [edit shared sae configuration aggregate-services]
   user@host# commit
   commit complete
   ```

Rolling Back Configuration Changes

This sample procedure shows how to use the rollback command to return to the most recently committed configuration. This command is useful if you make configuration changes, and then decide not to keep the changes.

This example shows how to view the default configuration for redirect server when the component is running, make configuration changes for redirect server, then return to the most recently committed configuration that does not include the changes. Redirect server redirects HTTP requests to a captive portal page.

1. Enter configuration mode.

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

2. View the current configuration (if any) for redirect server.

   ```
   [edit]
   user@host# show redirect-server
   tcp-port 8800;
   destination-url;
   refresh;
   refresh-document etc/refresh.html;
   request-rate 12000;
   request-burst-size 18000;
   client-rate 25;
   client-burst-size 50;
   ```

   The statements in the output show the default configuration for redirect server.

3. Configure the destination URL for redirect server.

   ```
   [edit]
   user@host# set redirect-server destination-url
   http://www.mycompany.com/default.html
   ```

4. View the updated configuration.

   ```
   [edit]
   user@host# show redirect-server
   tcp-port 8800;
   destination-url http://www.mycompany.com/default.html;
   refresh;
   refresh-document etc/refresh.html;
   request-rate 12000;
   request-burst-size 18000;
   client-rate 25;
   client-burst-size 50;
   ```

   The destination-url statement shows the new URL.

5. Use the rollback configuration mode command to return to the most recently committed configuration.

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

6. View the configuration again to make sure that your change is no longer present.

   ```
   [edit]
   user@host# show redirect-server
   tcp-port 8800;
   destination-url;
   refresh;
   refresh-document etc/refresh.html;
   request-rate 12000;
   request-burst-size 18000;
   client-rate 25;
   client-burst-size 50;
   ```

   The destination-url statement shows there is no URL.

7. Exit configuration mode.

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

Chapter 3

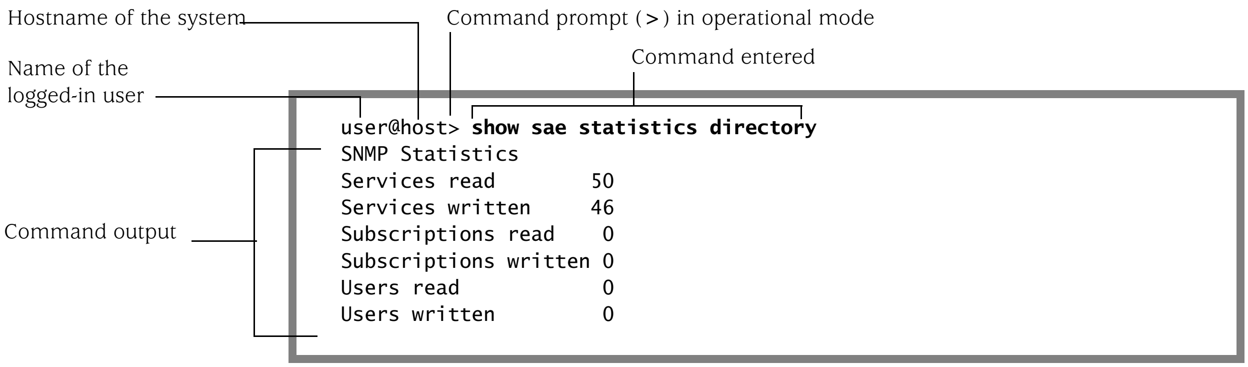
SRC CLI Basics

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

Elements of the Command-Line Interface

Figure 4 shows elements of the command-line interface in operational mode.

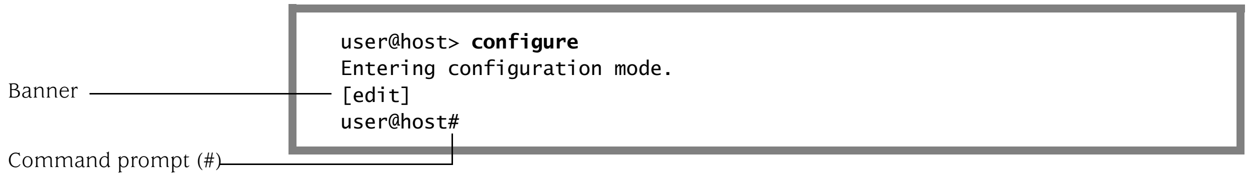
Figure 4: Elements of the Command-Line Interface


Hostname of the system
Command prompt (>) in operational mode
Command entered
Name of the logged-in user
Command output

```
user@host> show sae statistics directory
SNMP Statistics
Services read        50
Services written     46
Subscriptions read    0
Subscriptions written 0
Users read            0
Users written         0
```

Figure 5 shows elements of the command-line interface in configuration mode. In configuration mode, the prompt changes from a > to a #.

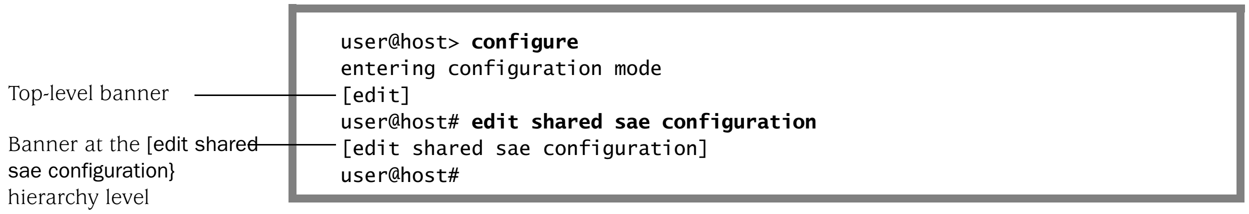
Figure 5: Command Prompt in Configuration Mode



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

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 6.)

Figure 6: Hierarchy-Level Banner


Top-level banner
Banner at the [edit shared sae configuration} hierarchy level

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

SRC CLI Messages

The CLI displays messages when you enter and exit from configuration and operational command modes, 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 sae <Enter>
                     ^
syntax error, expecting <command>.

[edit]
user@host# display

           ^
unknown command.
```

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 a slot number to complete the command:

```
user@host# edit slot
                   ^
syntax error, expecting <identifier>.
```

In this example, you need to type a value for the keepalive time to complete the command:

```
user@host# set shared sae configuration aggregate-services keepalive-time
                                                                    ^
syntax error, expecting <data>.
```

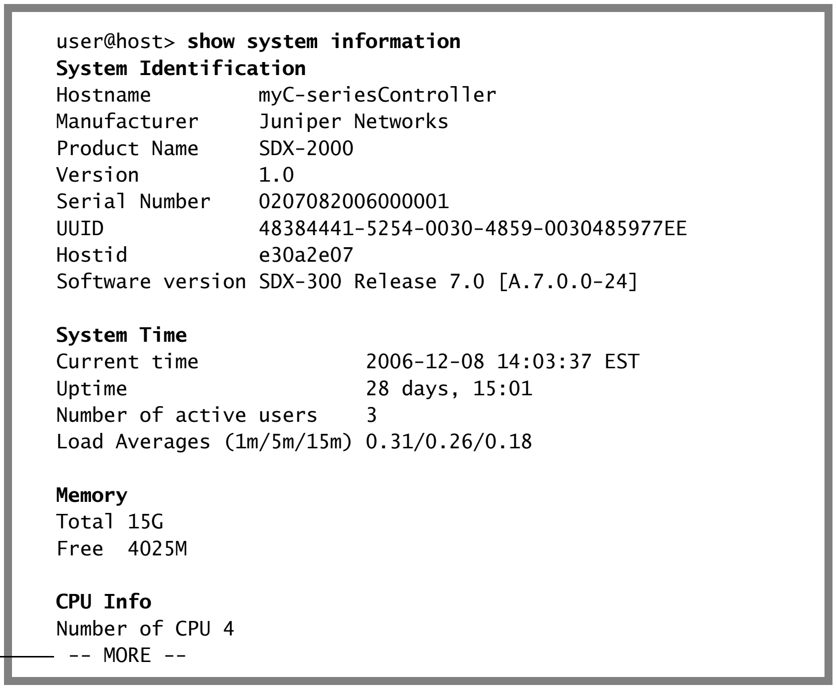
If you have omitted a required statement at a particular hierarchy level, when you attempt to move from that hierarchy level or when you issue the show command in configuration mode, a message indicates which statement is missing. For example:

```
[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 7.)

Figure 7: The MORE Prompt



```
user@host> show system information
System Identification
Hostname         myC-seriesController
Manufacturer     Juniper Networks
Product Name     SDX-2000
Version          1.0
Serial Number    0207082006000001
UUID             48384441-5254-0030-4859-0030485977EE
Hostid           e30a2e07
Software version SDX-300 Release 7.0 [A.7.0.0-24]

System Time
Current time              2006-12-08 14:03:37 EST
Uptime                    28 days, 15:01
Number of active users    3
Load Averages (1m/5m/15m) 0.31/0.26/0.18

Memory
Total 15G
Free  4025M

CPU Info
Number of CPU 4
 -- MORE --
```

The ---More--- prompt

To continue command output:

- Press Enter.

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

To cancel command output:

- Press q. Command output stops, and the command prompt appears.

Table 4 lists common keyboard sequences you can use at the —(more)— prompt.

Table 4: MORE Prompt Keyboard Sequences

Category	Action	Keyboard Sequence
Scroll down	Scroll down one line.	e, Ctrl+e, j, Ctrl+n, Enter, down arrow
	Scroll down one-half screen.	d, Ctrl+d
	Scroll down one whole screen.	f, Ctrl+f, Ctrl+v, z, Space
	Scroll down to the bottom of the output and wait for more input. (To resume output, press Ctrl-C.)	F
	Jump to last line in output and exit to the CLI prompt.	G
Scroll up	Display the previous line of output.	y, Ctrl+y, k, Ctrl+k, Ctrl+p, up arrow
	Scroll up one-half screen.	u, Ctrl+u
	Scroll up one whole screen.	b, Ctrl+b, Esc+v, w
	Jump to the first line of the output.	g
Scroll up and down	Scroll up and down through the output. To exit this mode, press q.	-E (hyphen E)

For more information about working with command output, see *Chapter 7, Filtering Command Output in the SRC CLI*.

Types of SRC Commands and Statements

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

- Operational mode commands—Commands that you enter in operational mode are used to monitor system operation.
    - For more information about using top-level CLI operational mode commands, see *Chapter 5, Using the SRC CLI Operational Commands to Monitor the SRC Software*.
    - For a complete list of CLI operational mode commands, see the *SRC-PE CLI Command Reference*.
- Environment commands—A set of operational mode commands that you can use to control the CLI environment. For example, you can specify editing level for the CLI. For more information, see *Chapter 9, Controlling the SRC CLI Environment*.

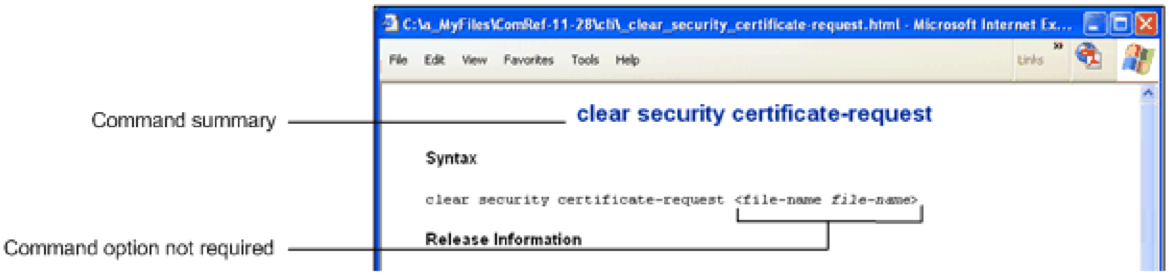
- Configuration mode commands—Commands that you enter in configuration mode are used to perform general configuration functions; for example, committing a configuration, navigating the hierarchy, and managing configuration files. For more information, see *Chapter 6, Using Commands and Statements to Configure the SRC Software*.
- Configuration statements—Used to define your SRC configuration. Your location in the configuration hierarchy determines which configuration statements are available. For example, the [edit shared sae] hierarchy level includes statements to configure the SAE.
  - For information about using SRC configuration statements, see the other SRC guides.
  - For a complete list of the SRC configuration statements, see the *SRC-PE CLI Command Reference*.

Command Options

When working on the command line, you are bound by specific CLI 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. Some commands may have options that are not required, allowing you to change the way the commands run or the information they return.

The command and statement summaries in the *SRC-PE CLI Command Reference* show which options are required and which options are not. Options at the top statement level that are not required are shown with angle brackets (< >). (See Figure 8.) Dots after an option indicate that more than one value can be supplied for the option.

Figure 8: Command Options


C:\a_MyFiles\ComRef-11-28\cli_clear_security_certificate-request.html - Microsoft Internet Ex...
File
Edit
View
Favorites
Tools
Help
Links
Command summary
Command option not required

clear security certificate-request

Syntax

```
clear security certificate-request <file-name file-name>
```

Release Information

Configuration Statements and Identifiers

You configure SRC 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:

```
[edit]
user@host# set ?
Possible completions:
> interfaces           Interfaces on the SDX platform
> policies             Policy configuration
> redirect-server      Redirect server properties
> routing-options      Protocol-independent routing option configuration
> services             Service configuration
> shared               Shared component information
> slot                 Component configuration
> snmp                 SNMP agent
> subscribers          Subscriber and subscription configuration
> system               System configuration
```

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.

The following list shows the statements available at the [edit system ntp] level of configuration mode. This level includes output that shows:

- No angle bracket (>) before the statement name—A leaf statement. You cannot define other statements at hierarchy levels below it.
- Plus sign (+) before the statement name—A statement that can contain a set of values. To specify a set, include the values in brackets.

```
[edit system ntp]
user@host# set ?
Possible completions:
> authentication-key   Configure NTP authentication keys
  boot-server          Server to query during boot sequence
> broadcast            Configure for broadcast mode
  broadcast-client     Listen for NTP broadcasts
> multicast-client     Listen for NTP multicasts
> peer                 NTP peer properties
> server               NTP server properties
+ trusted-key          List of trusted authentication keys (1..INF)
```

Listings can also include:

- Asterisk (*) before a statement name—A required statement or option that is not configured.
- Asterisk and Plus (+ *) before a statement name—Required options that can contain a set of values.

When you type a statement, enclose in quotation marks (double quotes) identifiers and any strings that include the following characters: space tab ( ) [ ] { } ! @ # $ % ^ & | ’ = ?

Privilege Levels for Using Commands and Statements

Each 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 **network** permissions can access the network by using the **telnet** and **shh** commands. The root login account has superuser privileges—with access to all commands and statements.

NOTE: Although `root` has superuser privileges, the editing level for `root` is set to normal.

For information about the editing level, see *Chapter 9, Controlling the SRC CLI Environment*.

Required privilege levels are listed in command and statement summaries. For more information about setting user accounts and privileges, see the *SRC-PE Getting Started Guide, Chapter 24, Configuring User Access with the SRC CLI*.

Switching Between Operational Mode and Configuration Mode

When you monitor and configure the SRC software, you may need to switch between operational mode and configuration mode. When you change to configuration mode, the command prompt also changes. By default, the operational mode prompt is a right angle bracket (>), and the configuration mode prompt is a pound sign (#).

Switching to Configuration Mode

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]
  user@host#
  ```

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

Returning to Operational Mode

You can return to operational mode with or without committing configuration changes. You can enter or exit configuration mode as many times as you wish without committing your changes.

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>
```

If there are changes that have not been committed, the CLI returns a message to that effect:

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

To return to operational mode from any configuration hierarchy level, such as [edit system services]:

```
[edit system services]
user@host# exit configuration-mode

user@host>
```

Running Operational Mode Commands from Configuration Mode

To display the output of an operational mode command, such as show, while in configuration mode:

- Issue the run configuration mode command, then specify the operational mode command.

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

  For example, to display the SAE configuration and then check whether the SAE is running:

  ```
  [edit shared sae configuration]
  user@host# show
  plug-ins {
    pool rksPlugin {
      pcmm-rks {
        load-balancing-mode failover;
  ```

```
          failback-timer -1;
          retry-interval 3000;
          maximum-queue-length 10000;
          feid-mso-domain-name abcd.com;
          trusted-element;
          default-peer radius01;
          peer-group rksPeer {
            server-address 10.10.3.60;
            server-port 1812;
          }
        }
      }
    }
    driver {
    . . .

    [edit shared sae configuration]
    user@host# run show component
    Installed Components
    Name      Version                                         Status
    cli       Release: 7.0 Build: CLI.A.7.0.0.0171            running
    acp       Release: 7.0 Build: ACP.A.7.0.0.0174            disabled
    jdb       Release: 7.0 Build: DIRXA.A.7.0.0.0176          running
    editor    Release: 7.0 Build: EDITOR.A.7.0.0.0176         disabled
    redir     Release: 7.0 Build: REDIR.A.7.0.0.0176          disabled
    licSvr    Release: 7.0 Build: LICSVR.A.7.0.0.0179         stopped
    nic       Release: 7.0 Build: GATEWAY.A.7.0.0.0170        disabled
    sae       Release: 7.0 Build: SAE.A.7.0.0.0166            running
    www       Release: 7.0 Build: UMC.A.7.0.0.0169            disabled
    jps       Release: 7.0 Build: JPS.A.7.0.0.0172            disabled
    agent     Release: 7.0 Build: SYSMAN.A.7.0.0.0174         disabled
    webadm    Release: 7.0 Build: WEBADM.A.7.0.0.0173         disabled
```

Moving Among Hierarchy Levels in the SRC CLI

You can use the CLI commands in Table 5 to navigate the levels of the configuration statement hierarchy.

Table 5: CLI Configuration Mode Navigation Commands

Command	Description
edit *hierarchy-level*	Moves to an existing configuration statement hierarchy or creates a hierarchy and moves to that level.
exit	Moves up the hierarchy to the previous level where you were working. This command is, in effect, the opposite of the edit command. Alternatively, you can use the quit command. exit and quit are interchangeable.
up	Moves up the hierarchy one level at a time.
top	Moves directly to the top level of the hierarchy.

Displaying SRC CLI Command History

To display a list of recent commands issued:

- In operational mode or configuration mode, use the history command.

```
user@host> history
469 show shared sae configuration ldap subscriber-data
  edit shared sae configuration ldap subscriber-data
  set subscription-loading-filter subscriberRefFilter
  set load-subscriber-schedules
  set login-cache-dn
  set session-cache-dn
  set server-address 10.10.10.3
  set dn umc=user
  set password abcde
  set directory-eventing
  set polling-interval 60
  set ldaps
  show
  set login-cache-dn o=Users,<base>
  . . .
```

You can copy a set of commands from the history and paste them into the CLI to execute the commands again.

CLI Support for C-series Controllers and for SRC Software Installed on Solaris Platforms

The CLI lets you manage and monitor a C-series Controller and configure and monitor the SRC software whether in runs on a C-series Controller or a Solaris platform. When the CLI runs on a C-series Controller, it provides system-level commands not available on a Solaris platform.

The CLI lets you perform the following system-level tasks on a C-series Controller:

- Configure and monitor:
  - Juniper Networks database
  - Platform interfaces that provide connectivity from the network to a platform
  - System log server
  - Network time Protocol (NTP)
- Configure system hostname, domain name service, security certificates, incoming SSH connections, incoming Telnet connections, RADIUS and TACACS+ authentication for users who access a C-series Controller
- Manage the system including system disks
- Install, upgrade, and uninstall system software
- View statistics for the Iptables tool

The CLI lets you perform the following tasks on a C-series Controller or for the SRC software installed on a Solaris platform:

- Configure and monitor:
    - ACP
    - JPS
    - NIC
    - Redirect server
    - SAE
    - SNMP agent
- Configure policies, services, subscribers, and subscriptions
- Manage security certificates
- Set the system date through the **set date** command
- View NTP status
- Manage files
- Discover all manageable network elements on a subnet
- Manage CLI settings
- Manage configurations
- Navigate in the CLI
- Work with files
- Configure user access

Chapter 4

Getting Online Help for the SRC CLI

This chapter describes the SRC CLI online help. Topics include:

Getting Help for Commands and Statements for the SRC CLI

Information about commands and statements is provided at each level of the CLI hierarchy. You can type a question mark to get help in the following ways:

- If you type the question mark at the command-line prompt, the CLI lists the available commands, statements, or 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 system-level objects
  configure              Enter configuration mode
  disable                Stop an SDX component
  enable                 Start an installed and configured component
  exit                   Exit a CLI session
  file                   Perform file operations
  help                   Display help about commands and statements
  history                Display command history
  request                Make request for specified component
  restart                Restart an SDX component
  set                    Properties for the CLI environment
  show                   Display configuration information
  ssh                    Open SSH session to another host
  start                  Start shell
  telnet                 Telnet to another host
  test                   Test a NIC resolution
user@host>
```

- If you type the question mark after entering the complete name of a command, statement, or option, the CLI lists the available options, then displays again the text that you typed.

```
user@host> clear ?
Possible completions:
  sae                     Clear SAE-level objects
  security                Delete security data
user@host> clear
```

- If you type the question mark in the middle of a command, statement, or option name, the CLI lists possible command completions that match the letters you have entered so far, then displays again the text 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 (delete) data
  configure            Enter configuration mode
user@host> c
```

You can also get extended help for a command, statement, or option by using the help command:

```
user@host> help enable
Start a specified SDX component.
```

Getting Help for Omitted Statements

If you omit a required statement at a particular hierarchy level, when you issue the show command in configuration mode, a message indicates which statement is missing. For example:

```
user@host# show system
user test {
    class subscriber-control; ## Warning: Undefined class
'subscriber-control'
    full-name test;
    uid 503;
    gid 100;
    authentication {
      encrypted-password "{crypt}0N5Od4BqvwPKY";
    }
```

Using SRC 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 Spacebar. If the partially typed letters begin a string that uniquely identifies a command, the complete command name appears. Otherwise, a prompt 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 a show command for a component that starts with the letter s:

```
user@host> show s
                  ^
's' is ambiguous.
Possible completions:
  sae                  Display SAE information
  security             Display security information
  system               Display system information
user@host>
```

Display a list of all log files whose names start with the letter m:

```
user@host> file show /var/log/m?
Possible completions:
  <filename>            Filename to show
  /var/log/maillog      Size: 0, Last changed: Sep 27, 2006 10:33 AM
  /var/log/messages     Size: 109569, Last changed: Oct 2, 2006 3:10 PM
```

Using Command Completion in Configuration Mode

The CLI command completion functions also apply to the commands in configuration mode and to configuration statements. 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 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:

```
[edit]
user@host# ?
Possible completions:
  commit              Commit a set of changes
  delete              Delete a configuration statement or identifier
  edit                Specify edit level in hierarchy
  exit                Exit from this level
  help                Display help about commands and statements
  history             Display command history
  insert              Insert an identifier
  load                Load configuration from an ASCII file
  rename              Rename a statement or identifier
  rollback            Discard current set of changes
  run                 Run an operational mode command
  save                Save configuration to a file
  set                 Set a configuration property
  show                Display configuration information
  top                 Return to top level of configuration mode
  up                  Move up one level in hierarchy
```

List all the statements available at the [edit] hierarchy level:

```
[edit]
user@host# edit ?
Possible completions:
> interfaces          Interfaces on the SDX platform
> policies            Policy configuration
> redirect-server     Redirect server properties
> routing-options     Protocol-independent routing option configuration
> services            Service configuration
> shared              Configure a shared configuration
> slot                Component configuration
> snmp                SNMP agent
> subscribers         Subscriber and subscription configuration
> system              System parameters
```

List all the statements available at the [edit system] hierarchy level:

```
[edit]
user@host# edit system ?
Possible completions:
  <[Enter]>           Execute this command
> ldap                LDAP properties
> login               Login properties
> ntp                 NTP configuration
> radius-server       RADIUS server configuration
> services            System services configuration
> syslog              System log configuration
> tacplus-server      TACACS+ server configuration
  |                   Pipe through a command
```

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

```
[edit]
user@host# edit system l?
Possible completions:
> ldap                LDAP properties
> login               Login properties
user@host# edit system l
```

List all configured Ethernet interfaces:

```
[edit]
user@host# edit interfaces et?
Possible completions:
  <name>              Interface name
  eth0
  eth1
  eth2
  eth3
user@host# edit interfaces et
```

Display a list of all configured interfaces:

```
user@host# show interfaces ?
Possible completions:
  <[Enter]>           Execute this command
  <name>              Interface name
  eth0                Interface name
  lo                  Interface name
  |                   Pipe through a command
user@host# show interfaces
```

Part 2

Operational Mode and Configuration Mode

Chapter 5

Using the SRC CLI Operational Commands to Monitor the SRC Software

This chapter provides information about CLI operational commands. Topics include:

SRC CLI Command Categories

When you log in to the SRC CLI and it starts, broad groups of CLI commands are available:

- Commands for controlling the CLI environment—The commands in the **set** hierarchy configure the CLI display screen.
- Commands for monitoring and troubleshooting—The following commands display information and statistics about the software and test network connectivity:
  - **clear**—Clears statistics and protocol database information.
  - **show**—Displays the current configuration and information about interfaces, routing protocols, routing tables, routing policy filters, system alarms, and the chassis.
  - **test**—Tests the configuration.
- 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.

- Commands for copying files—The **file copy** command copies files from one location on the system to another, from the system to a remote system, or from a remote system to the local system.
- Commands for restarting software processes—The commands in the **restart** hierarchy restart the various SRC components, including the Admission Control Plug-In (ACP), SNMP agent, the service and policy editor, the Juniper Networks database, Juniper Policy Server, the network information collector, the redirect server, SAE, and C-Web.
- A command—**request**—for performing system-level operations, including stopping and rebooting the C-series Controller and loading SRC software images.
- A command—**start**—to exit the CLI and start a shell.
- A command—**configure**—to enter configuration mode, which provides a series of commands that configure the SRC software, including system management and SRC components.
- A command—**quit**—to exit the CLI.

For more information about the CLI operational mode commands, see *SRC-PE CLI Command Reference*.

Commonly Used Operational Mode Commands

Table 6 lists operational commands that you may find useful to monitor system operation. For a complete description of operational commands, see *SRC-PE CLI Command Reference*.

Table 6: Commonly Used Operational Mode Commands

Items to Check	Description	Command
CLI	Settings for CLI environment	show cli
Configuration	Current system configuration	show configuration
Manipulate files	List of files and directories	file list
	Contents of a file	file show
Redirect server	Redirect server usage information	show redirect-server statistics
SRC licenses	Type of license and detailed license information	show sae licenses
System	General system information, including hostname, hardware version, software version, disk usage	show system information
System components	List of installed components and the status of each component	show component

Viewing Files and Directories

The SRC software stores information in files on the system, including configuration files, log files, and system software files. You can use operational commands to view files and directories on the system.

Directories on the C-series Controller

The C-series Controller has numerous directories used by the operating system. Table 7 lists directories on a C-series Controller.

Table 7: Directories on a C-series Controller

Directory	Description
/altroot	Files that are a snapshot of the file system. You can use these files to restore the file system to the state in the snapshot.
/altvar	Files that are a snapshot of the file system. You can use these files to restore the file system to the state in the snapshot.
/media	Mount points created automatically for dynamic devices (for example, USB flash drive)
/opt/UMC	SRC files for installed components
/tmp	Temporary files
/var/home	Home directory for local users
/var/log	System log files
/var/UMC	Operational files and log files for SRC components

Listing Files and Directories

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

The user's home directory is the default directory for most of the SRC software commands that require a filename.

NOTE: You can change the default directory by using the **set cli directory** command.

- To view a list of the **file** commands, type the following:

```
user@host> file ?
Possible completions:
  archive               Archive files from the system (local)
  checksum              Calculate file checksum
  compare               Compare files (local)
  copy                  Copy files
  delete                Delete a file (local)
  list                  List files (local)
  rename                Rename a file (local)
  show                  Show file contents
user@host> file
```

- Use the list option to see the directory structure. For example, to show the files located in your home directory:

```
user@host> file list
initial.cfg
install.log
install.log.syslog
```

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

```
user@host> file list /opt/UMC/
acp
agent
cli
editor
idp
jdb
jps
jre
licsvr
net-snmp
nic
pom
redir
sae
smg
webadm
```

- You can also use the CLI context-sensitive help system to locate a directory. For example:

```
user@host> file list /?
Possible completions:
  <path>                Path to list
  /.autofsck            Size: 0, Last changed: Sep 19, 2006 1:36 PM
  /altroot/             Last changed: Sep 19, 2006 1:22 PM
  /bin/                 Last changed: Sep 19, 2006 1:27 PM
  /boot/                Last changed: Sep 19, 2006 1:25 PM
  /dev/                 Last changed: Sep 19, 2006 1:36 PM
  /etc/                 Last changed: Sep 19, 2006 2:42 PM
  /home/                Last changed: Feb 18, 2005 4:26 AM
  /initrd/              Last changed: Feb 18, 2005 4:26 AM
  /lib/                 Last changed: Sep 19, 2006 1:26 PM
  /lib64/               Last changed: Sep 19, 2006 1:26 PM
  /lost+found/          Last changed: Sep 19, 2006 1:22 PM
  /media/               Last changed: Sep 19, 2006 1:36 PM
  /misc/                Last changed: Aug 15, 2006 8:33 PM
  /mnt/                 Last changed: Feb 18, 2005 4:26 AM
  /net/                 Last changed: Sep 19, 2006 1:36 PM
  /opt/                 Last changed: Sep 19, 2006 1:27 PM
  /proc/                Last changed: Sep 19, 2006 1:36 PM
  /root/                Last changed: Sep 19, 2006 6:44 PM
  /sbin/                Last changed: Sep 19, 2006 1:26 PM
  /selinux/             Last changed: Sep 19, 2006 1:36 PM
  /srv/                 Last changed: Feb 18, 2005 4:26 AM
  /sys/                 Last changed: Sep 19, 2006 1:36 PM
  /tmp/                 Last changed: Sep 19, 2006 6:46 PM
  /usr/                 Last changed: Sep 19, 2006 1:24 PM
  /var/                 Last changed: Sep 19, 2006 1:36 PM
```

```
user@host> file list /var/?
Possible completions:
  <path>                     Path to list
  /var/UMC/                  Last changed: Sep 19, 2006 1:28 PM
  /var/account/              Last changed: Sep 19, 2006 1:25 PM
  /var/acp/                  Last changed: Sep 19, 2006 1:27 PM
  /var/cache/                Last changed: Sep 19, 2006 1:26 PM
  /var/crash/                Last changed: Sep 19, 2006 1:25 PM
  /var/db/                   Last changed: Sep 19, 2006 1:26 PM
  /var/empty/                Last changed: Sep 19, 2006 1:26 PM
  /var/home/                 Last changed: Sep 19, 2006 6:44 PM
  /var/lib/                  Last changed: Sep 19, 2006 1:36 PM
  /var/local/                Last changed: Feb 18, 2005 4:26 AM
  /var/lock/                 Last changed: Sep 19, 2006 1:36 PM
  /var/log/                  Last changed: Sep 19, 2006 2:41 PM
  /var/lost+found/           Last changed: Sep 19, 2006 1:22 PM
  /var/mail/                 Last changed: Feb 18, 2005 4:26 AM
  /var/net-snmp/             Last changed: Sep 19, 2006 6:44 PM
  /var/nis/                  Last changed: Feb 18, 2005 4:26 AM
  /var/opt/                  Last changed: Feb 18, 2005 4:26 AM
  /var/preserve/             Last changed: Feb 18, 2005 4:26 AM
  /var/run/                  Last changed: Sep 19, 2006 6:44 PM
  /var/spool/                Last changed: Sep 19, 2006 1:26 PM
  /var/tmp/                  Last changed: Sep 19, 2006 1:28 PM
  /var/yp/                   Last changed: Sep 19, 2006 1:26 PM
```

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

```
user@host> file show install.log
Installing 309 packages

Installing chkconfig-1.3.13.4-1.x86_64.
Installing ethtool-1.8-4.x86_64.
Installing hdparm-5.7-2.x86_64.
Installing hwdata-0.146.22.EL-1.noarch.
Installing iputils-20020927-18.EL4.3.x86_64.
Installing libgcc-3.4.6-3.i386.
Installing libgcc-3.4.6-3.x86_64.
Installing mingetty-1.07-3.x86_64.
Installing mktemp-1.5-20.x86_64.
Installing redhat-logos-1.1.26-1.junosx.noarch.
Installing rootfiles-8-1.noarch.
Installing setserial-2.17-17.x86_64.
. . .
```

Specifying Filenames and URLs

In some CLI commands and configuration statements—including file copy, file archive, load, and save—you can include a filename.

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

- *filename*—File in the user's current directory on the local system. You can use wildcards to specify multiple source files or a single destination file. Wildcards are not supported in FTP.

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

- *path/filename*—File on the local system.
- *File URL*—File URL of local files.
- usb:*filename*—Files on a dynamically mounted USB port.
- ftp://*hostname/path/filename*—File on an FTP server. You can also specify *hostname* as *username@hostname* or *username*:*password@hostname*.

  For example:

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

Managing SRC Components

You use operational commands to manage SRC components.

Viewing C-series Controller Information

You can view general information about the C-series Controller and version and status information for the installed SRC components.

To view SRC software information, including hostname and version information for the SRC software installed on your system, type the following command:

```
user@host> show system information
System Identification
Hostname                    myC-seriesController
Manufacturer                Juniper Networks
Product Name                SDX-2000
Version                     1.0
Serial Number               0207082006000001
UUID                        48384441-5254-0030-4859-0030485977EE
Hostid                      e30a2e07
Software version SDX-300 Release 7.0 [A.7.0.0-24]

System Time
Current time                                        2006-11-30 12:09:15 EST
Uptime                                              20 days, 13:06
Number of active users                              4
Load Averages (1m/5m/15m)                               0.19/0.22/0.18

Memory
Total   15G
Free    6955M

CPU Info
Number of CPU               4
CPU Model                   Dual Core AMD Opteron(tm) Processor 265
Clock Speed                 1804.158 MHz

Disk Information
  Mountpoint       Total Used  Use%
  /                         2015M 1310M  64%
  /altroot                  2015M   35M    1%
  /altvar                       29G   75M   0%
  /boot                         98M   14M  14%
  /var                          31G 1850M   5%

Temperature
System +27 C
CPU-1  +37 C
CPU-2  +40 C

Fan Speed
Fan-1 9375 RPM
Fan-2 9375 RPM
```

- To display version and status information for each component installed on the C-series Controller, in operational mode type the following command:

```
user@host> show component
Installed Components
Name      Version                                            Status
cli       Release: 7.0 Build: CLI.A.7.0.0.0171               running
acp       Release: 7.0 Build: ACP.A.7.0.0.0174               disabled
jdb       Release: 7.0 Build: DIRXA.A.7.0.0.0176             running
editor    Release: 7.0 Build: EDITOR.A.7.0.0.0176            disabled
redir     Release: 7.0 Build: REDIR.A.7.0.0.0176             disabled
```

```
licSvr    Release: 7.0 Build: LICSVR.A.7.0.0.0179           stopped
nic       Release: 7.0 Build: GATEWAY.A.7.0.0.0170          disabled
sae       Release: 7.0 Build: SAE.A.7.0.0.0166              running
www       Release: 7.0 Build: UMC.A.7.0.0.0169              disabled
jps       Release: 7.0 Build: JPS.A.7.0.0.0172              disabled
agent     Release: 7.0 Build: SYSMAN.A.7.0.0.0174           disabled
webadm    Release: 7.0 Build: WEBADM.A.7.0.0.0173           disabled
```

Restarting an SRC Component

In some instances, you may need to restart a SRC component. You can restart a component gracefully, immediately, or by sending a hangup signal before restarting the component. Table 8 shows options available for the restart command.

Table 8: Options to Restart an SRC Component

Option	Description
component	Name of SRC component to restart.
gracefully *component*	Restart a specified SRC component by sending the equivalent of a UNIX SIGTERM signal.
immediately *component*	Immediately restart an SRC component by sending the equivalent of a UNIX SIGKILL signal.
soft *component*	Reread and reactivate the configuration without completely restarting the SRC component. This option is the equivalent of a UNIX SIGHUP signal; omitting this option is the equivalent of a UNIX SIGTERM (kill) operation.

To restart a component:

- In operational mode, enter a restart command.

  For example, to gracefully restart the NIC component:

  ```
  user@host> restart gracefully component UMCnic
  Shutting down the NICHOST server: done
  Starting NICHOST: may take a few minutes...
  ```

Stopping the SRC Software

To gracefully shut down the SRC software and power off the system, in operational mode, enter the following command:

```
user@host> request system halt
```

This command stops all system components, halts the operating system, and powers down the C-series Controller.

NOTE: The request system halt command does not give you the opportunity to restart the system from the CLI.

For example:

```
user@host> request system halt
Halt the system [yes,no] ? (no) y
BroadcaStopping HAL daemon: [  OK  ]
Stopping system message bus: [  OK  ]
Stopping atd: [  OK  ]
Shutting down xfs: [  OK  ]
Stopping sshd:[  OK  ]
Shutting down smartd: [  OK  ]
Stopping snmpd: [  OK  ]
Stopping xinetd: [  OK  ]
Stopping acpi daemon: [  OK  ]
Stopping crond: [  OK  ]
Stopping autofs:  [  OK  ]
Stopping nscd: [  OK  ]
Shutting down ntpd: [  OK  ]
Stopping NFS statd: [  OK  ]
Stopping irqbalance: [  OK  ]
Stopping portmap: [  OK  ]
Shutting down kernel logger: [  OK  ]
Shutting down system logger: [  OK  ]
Stopping pcmcia:  unloading Kernel Card Services
[  OK  ]
Stopping sysstat:  [  OK  ]
Starting killall:  [  OK  ]
Sending all processes the TERM signal...
Sending all processes the KILL signal...
Saving random seed:
Syncing hardware clock to system time
Turning off swap:
Turning off quotas:
Unmounting pipe file systems:
Unmounting file systems:
Halting system...
md: stopping all md devices.
md: md0 switched to read-only mode.
Synchronizing SCSI cache for disk sda:
Power down.
acpi_power_off called
```

Rebooting the SRC Software

In some instances, such as after software upgrades that make changes to the operating system kernel, you need to reboot the SRC software. Reboot requests are recorded to the system log files and the messages about the final stages of system appear on the screen when the command to reboot the system is run.

To reboot the SRC software, enter the following command in operational mode:

```
user@host> request system reboot

Reboot the system [yes,no] ? (no) y
BroadcasStopping HAL daemon: [  OK  ]
Stopping system message bus: [  OK  ]
Stopping atd: [  OK  ]
Shutting down xfs: [  OK  ]
Stopping sshd:[  OK  ]
Shutting down smartd: [  OK  ]
```

```
Stopping snmpd: [  OK  ]
Stopping xinetd: [  OK  ]
Stopping acpi daemon: [  OK  ]
Stopping crond: [  OK  ]
Stopping autofs:  [  OK  ]
Stopping nscd: [  OK  ]
Shutting down ntpd: [  OK  ]
Stopping NFS statd: [  OK  ]
Stopping irqbalance: [  OK  ]
Stopping portmap: [  OK  ]
Shutting down kernel logger: [  OK  ]
Shutting down system logger: [  OK  ]
Stopping pcmcia:  unloading Kernel Card Services
[  OK  ]
Stopping sysstat:  [  OK  ]
Starting killall:  [  OK  ]
Sending all processes the TERM signal...
Sending all processes the KILL signal...
Saving random seed:
Syncing hardware clock to system time
Turning off swap:
Turning off quotas:
Unmounting pipe file systems:
Unmounting file systems:
Please stand by while rebooting the system...
md: stopping all md devices.
md: md0 switched to read-only mode.
Synchronizing SCSI cache for disk sda:
Restarting system.
```

Viewing Information about Users Logged Into the SRC Software

You can get information about users currently logged into the SRC software form the SRC CLI.

To get information about users who are logged in:

```
user@host> show system users
  4:41pm  up 3 day(s),  1:01,  0 users,  load average: 0.04, 0.04, 0.12
User     tty          login@  idle   JCPU   PCPU  what
admin1  pts/6        9:30am    12      9      5   cli
user2   pts/10       4:40pm                       cli
admin1  pts/3        1:25pm  1:19                 telnet server2
```

where

- `User`—Specifies the name of the user logged in to the SRC software.
- `tty`—Specifies the terminal used for the user's connection.
- `login@`—Specifies the time at which the user logged in tot he SRC software.
- `idle`—Specifies how long the connection the connection has not had any activity.
- `JCPU`—Specifies the length of time used by processes for this terminal, including processes currently running in the background.
- `PCPU`—Specifies the time used by the process specified in the `what` field.
- `what`—Specifies the name of the processes currently in use by the specified user.

Chapter 6

Using Commands and Statements to Configure the SRC Software

This chapter describes how to use the CLI to configure the SRC software. Topics include:

Understanding SRC CLI Configuration Mode

In configuration mode, you can configure properties for the SRC software, such as properties for the Juniper Networks database, SRC components, user access, and system properties. You can configure all SRC properties for the C-series Controller, but only a subset of these properties if you are running SRC software on a Solaris platform.

A 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 statements, you commit them, which activates the configuration.

You can create the hierarchy interactively at the CLI, or you can a load configuration from a file that you create. To activate the configuration, you commit it.

For information about loading a configuration from a file, see *Chapter 10, Managing SRC Configurations*.

For information about CLI support for SRC installations on Solaris platforms, see *Chapter 3, SRC CLI Basics*.

Configuration Mode Commands

Table 9 summarizes each CLI configuration mode command. The commands are listed alphabetically.

Table 9: Summary of Configuration Mode Commands

Command	Description
commit	Commit the set of changes to the database and cause the changes to take operational effect.
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 before the last edit command, or exit from configuration mode. The quit and exit commands are synonyms.
help	Display help about available configuration statements.
history	Display the previous commands entered at the CLI.
insert	Insert an identifier into an existing hierarchy.
load	Load a configuration from a file. 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 before the last edit command, or exit from configuration mode. The quit and exit commands are synonyms.
rename	Rename an existing configuration statement or identifier.
rollback	Return to the previously committed configuration. The software saves only the last committed configuration.
run	Run an operational mode CLI command without exiting from configuration mode.
save	Save the configuration to a file in text or XML format. The contents of the current level of the statement hierarchy (and below) are saved, with the statement hierarchy containing it. This action allows a section of the configuration to be saved with the statement hierarchy.
set	Create a statement hierarchy and set identifier values. This command is similar to edit except that your current level in the hierarchy does not change.
show	Display the current 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.

For more information about configuration mode commands, see *SRC-PE CLI Command Reference*.

Configuration Statements

You configure SRC properties by including statements in the configuration. A statement consists of the following parts:

- Keyword—Fixed text
- Identifier (Optional)—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 10 describes top-level CLI configuration mode statements.

Table 10: Configuration Mode Top-Level Statements

Statement	Description
interfaces	Configure interfaces on the C-series Controller.
policies	Configure routing policies.
redirect-server	Configure the redirect server.
routing-options	Configure static routes.
services	Define subscriber services.
shared	Configure ACP, admission control, congestion points, auth-cache, network devices, NIC, and SAE.
slot	Configure properties for a component, such as ACP, Juniper Policy Server, network information collector, and SAE on a slot.
snmp	Configure Simple Network Management Protocol (SNMP) community strings, interfaces, traps, and notifications.
subscribers	Configure subscriber definitions.
system	Configure systemwide properties, including the hostname, domain name, Domain Name System (DNS) server, user logins and permissions, and software processes.

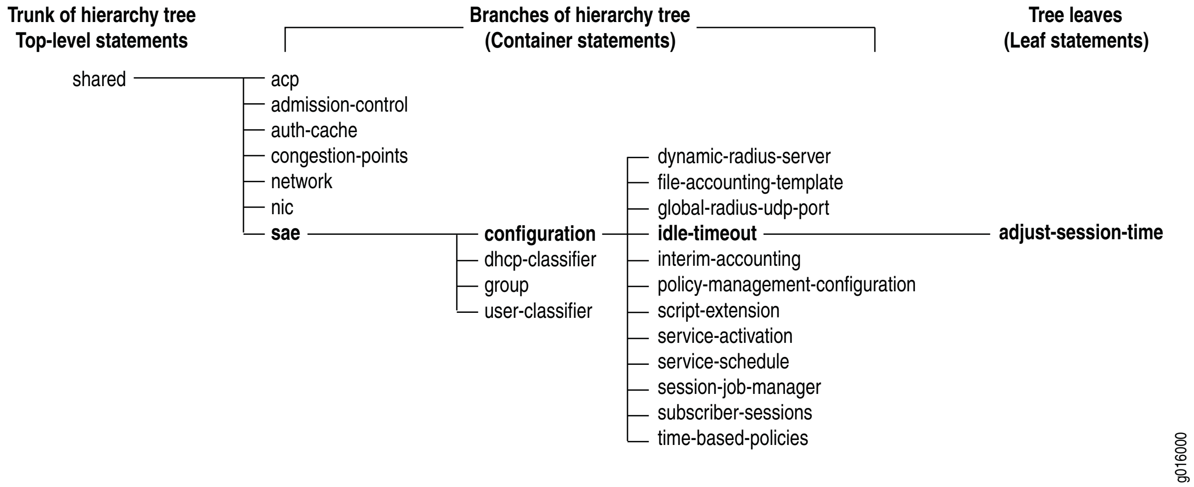
For information about specific configuration statements, see the *SRC-PE CLI Command Reference*.

Configuration Statement Hierarchy

The SRC 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. All the container and leaf statements together form the *configuration hierarchy*.

Figure 9 shows container statements and leaf statements in the **sae** hierarchy. To view this hierarchy at the CLI, the editing level must be set to expert.

Figure 9: Sample 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 9 illustrates the hierarchy tree, showing a statement path for the portion of the shared configuration hierarchy that configures the idle timeout for the SAE.

The **shared** statement is a top-level statement at the trunk of the configuration tree. The **acp**, **admission-control**, **auth-cache**, **congestion-points**, **network**, **nic**, and **sae** statements are all subordinate container statements of the **shared** statement (they are branches of the **shared** hierarchy tree).The **configuration** and the **idle-timeout** statements are successive branches in the hierarchy under the **sae** branch. The **adjust-session-time** statement is a leaf on the tree, which, in this case, specifies that when a session is terminated by an idle timeout, the session time reported in the accounting stop message is automatically reduced by the idle time.

The CLI represents the statement path shown in Figure 9 on page 54 as [**shared sae configuration idle-timeout**], and displays the configuration as follows:

```
shared {
    sae {
      configuration {
        idle-timeout {
          adjust-session-time;
        }
      }
    }
}
```

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. Each leaf statement ends with a semicolon.

Entering and Exiting Configuration Mode in the SRC CLI

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

Users must have configure permission to view and use the configure command to enter configuration mode. When in configuration mode, users can view and modify only those statements for which they have access privileges set.

Entering Configuration Mode

To enter configuration mode:

- Use the configure command.

```
user@host> configure
Entering configuration mode.
```

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

```
[edit]
user@host# ?
Possible completions:
    commit              Commit a set of changes
    delete              Delete a configuration statement or identifier
    edit                Specify edit level in hierarchy
    exit                Exit from this level
    help                Display help about commands and statements
    history             Display command history
    insert              Insert an identifier
    load                Load configuration from an ASCII file
    rename              Rename a statement or identifier
    rollback            Discard current set of changes
    run                 Run an operational mode command
    save                Save configuration to file
    set                 Set a configuration property
    show                Display configuration information
    top                 Return to top level of configuration mode
    up                  Move up one level in hierarchy
```

Exiting from Configuration Mode

To exit configuration mode:

- Use the exit configuration-mode command from any level.

  or

  Use the exit command from the top level.

For example:

```
[edit shared sae configuration time-based-policies]
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 by using the exit command and the configuration contains changes that have not been committed, you see a message:

```
[edit shared sae configuration time-based-policies]
user@host# exit configuration-mode
Exiting configuration mode.
The configuration has been changed but not committed.

user@host>
```

Modifying the Configuration

To configure the SRC software or to modify an existing 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>
  ```

  where *statement-path* is the hierarchy to the configuration statement and the statement itself.

- 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>
  ```

  where

  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.

Entering Values for Statement Options

When values include the following characters—space, single quotation marks (‘), double quotation marks (“), curly braces ({}) , brackets ([ ]), or commas (,)—you must enclose the value in quotation marks (“ “) or use a backslash (\) before the character.

To enter words or letters separated by a space, such as a full name with a first name and last name, enclose the words in quotation marks. For example:

```
"Chris Bee"
```

To enter multiple values, separate values with a space, and enclose the values with brackets. For example:

```
[192.0.2.24 192.0.4.25]
```

To enter a number using a regular expression, use backslashes (\) to escape the brackets. For example:

```
\[0-9\]
```

Displaying the Current Configuration

You can display the current configuration from operational mode or from configuration mode. In configuration mode, you can display the configuration at the specified hierarchy level.

To display the current configuration from configuration mode:

- Use the show command.

```
[edit]
user@host# show <statement-path>
```

or

```
[edit <hierarchy-level>]
user@host# show
```

To display the current configuration from operational mode:

- Use the show configuration command.

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

The configuration statements appear in a fixed order; however, when you configure the C-series Controller, you can enter statements in any order.

If you omit a required statement at a particular hierarchy level, when you issue the show or show configuration command, 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 or show configuration command.

For example, the following output includes a warning that lists mandatory attributes that need to be configured:

```
[edit]
user@host# show
. . .
shared {
  sae {
    configuration {
      aggregate-services {
        keepalive-time 172800;
        warning: missing mandatory attribute(s): 'keepalive-retry-time',
'activation-deactivation-time', 'failed-notification-retry-time'
      }
    }
   . . .
 }
}
```

Examples: Displaying the Current Configuration

Configure timers for aggregate services from the [edit] hierarchy level, and then view the configuration from the same hierarchy level:

```
[edit]
user@host# set shared sae configuration aggregate-services keepalive-time 86400
[edit]
user@host# set shared sae configuration aggregate-services keepalive-retry-time 900
[edit]
user@host# set shared sae configuration aggregate-services activation-deactivation-time 900
[edit]
user@host# set shared sae configuration aggregate-services failed-notification-retry-time 86400

[edit]
user@host# show
. . .
shared {
  sae {
    configuration {
      aggregate-services {
        keepalive-time 86400;
        keepalive-retry-time 900;
        activation-deactivation-time 900;
        failed-notification-retry-time 86400;
      }
    }
   . . .
  }
}
```

Display a configuration at a specific hierarchy level:

```
[edit]
user@host# show shared sae configuration aggregate-services
keepalive-time 172800;
keepalive-retry-time 900;
activation-deactivation-time 900;
failed-notification-retry-time 86400;
```

Move to a lower level in the hierarchy, [edit shared sae configuration aggregate-services], and then display the configuration at that level:

```
[edit]
user@host# edit shared sae configuration aggregate-services

[edit shared sae configuration aggregate-services]
user@host# show
keepalive-time 172800;
keepalive-retry-time 900;
activation-deactivation-time 900;
failed-notification-retry-time 86400;
```

Display all of the last committed configuration from operational mode:

```
[edit]
user@host# set shared sae configuration aggregate-services keepalive-time 172800
[edit]
user@host# set shared sae configuration aggregate-services keepalive-retry-time 900
[edit]
user@host# set shared sae configuration aggregate-services activation-deactivation-time 900
[edit]
user@host# set shared sae configuration aggregate-services failed-notification-retry-time 86400
[edit]
user@host# show
user@host# commit
commit complete.
[edit]
user@host# quit
Exiting configuration mode.

user@host> show configuration
. . .
shared {
  sae {
    configuration {
      aggregate-services {
        keepalive-time 172800;
        keepalive-retry-time 900;
        activation-deactivation-time 900;
        failed-notification-retry-time 86400;
      }
    }
   . . .
  }
}
```

Adding Configuration Statements and Identifiers

When you use the **?** help to view a list of possible command completions, the output includes symbols that provide more information about the statement. The following symbols can appear in a list:

- Angle bracket (>) before the statement name indicates that the statement is a container statement and that you can define other statements at levels below it.
- No angle bracket (>) before the statement name indicates that the statement is a leaf statement; you cannot define other statements at hierarchy levels below it.
- Plus sign (+) before the statement name indicates that the statement can contain a set of values. To specify a set, include the values in brackets.
- Asterisk (*) before a statement name indicates a required statement or option that is not configured.
- Plus/Asterisk (+ *) before a statement name indicates a required option that can contain a set of values.

The following example at the [edit system services] hierarchy level shows that authentication-order, domain-search, and name-server can contain more than one value.

```
[edit system]
user@host# show ?
Possible completions:
  <[Enter]>            Execute this command
+ authentication-order Order in which authentication methods are invoked
+ domain-search        List of domain names to search
  host-name            Hostname for SDX platform
> ldap                 LDAP properties
> login                Login properties
+ name-server          DNS name servers
> ntp                  NTP configuration
> radius-server        RADIUS server configuration
> services             System services configuration
> syslog               System log configuration
> tacplus-server       TACACS+ server configuration
  time-zone            Time zone definition name
  |                    Pipe through a command
```

The following example at the [edit shared sae configuration driver] hierarchy level shows that mac-cache-expiration and unauthenticated-subscriber-dn are required statements.

```
[edit shared sae configuration driver]
user@host# show ?
Possible completions:
  <[Enter]>            Execute this command
> junos                Parameters the SAE uses to manage JUNOS routing platforms
> junose               Parameters the SAE uses to manage JUNOSe routers
* mac-cache-expiration Time that a subscriber profile remains in SAE cache (0..I
NF s)
> pcmm                 Parameters that SAE uses to manage PCMM devices
> scripts              Scripts to customize setup of SAE connections to devices
> simulated            Parameters that the SAE uses for simulated drivers
> snmp                 Global default SNMP communities
> third-party          Parameters that SAE uses to manage third-party devices
* unauthenticated-subscriber-dn
                       DN of an unauthenticated subscriber profile
  virtual-portal-address
                       IP address of the portal server
  |                    Pipe through a command
```

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 as an identifier to complete the logger name:

```
[edit]
user@host# set shared sae configuration logger
                                               ^
syntax error, expecting <identifier>.
```

Deleting a Statement from the Configuration

Deleting a statement or an identifier effectively “unconfigures” the functionality associated with that statement or identifier, returning that functionality to its default condition.

To delete a statement or identifier:

- Use the **delete** configuration mode command.

  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:

- In configuration mode, use the **delete** command. Do not specify a statement or an identifier.

  When you omit the statement or identifier, you are prompted to confirm the deletion. For example:

  ```
  [edit]
  user@host# delete
  Delete everything under this level? [yes, no] (no)?
  ```

Examples: Deleting a Statement from the Configuration

Configure the aggregate-services statements, then delete these statements from the [edit] level. Using the delete command effectively unconfigures the SAE properties for aggregate-services in the SRC software:

```
[edit]
user@host# set shared sae configuration aggregate-services keepalive-time 172800
[edit]
user@host# set shared sae configuration aggregate-services keepalive-retry-time 900
[edit]
user@host# set shared sae configuration aggregate-services activation-deactivation-time 900
[edit]
user@host# set shared sae configuration aggregate-services failed-notification-retry-time 86400

[edit]
user@host# show
. . .
shared {
  sae {
    configuration {
      aggregate-services {
        keepalive-time 172800;
        keepalive-retry-time 900;
        activation-deactivation-time 900;
        failed-notification-retry-time 86400;
      }
    }
   . . .
  }
}

[edit]
user@host# delete shared sae configuration aggregate-services

[edit]
user@host# show shared sae configuration aggregate-services
```

Configure the aggregate-services statements, then delete these statements from the [edit shared sae configuration aggregate-services] level:

```
[edit]
user@host# set shared sae configuration aggregate-services keepalive-time 172800
[edit]
user@host# set shared sae configuration aggregate-services keepalive-retry-time 900
[edit]
user@host# set shared sae configuration aggregate-services activation-deactivation-time 900
[edit]
user@host# set shared sae configuration aggregate-services failed-notification-retry-time 86400
[edit]
user@host# edit shared sae configuration aggregate-services

[edit shared sae configuration aggregate-services]
user@host# delete
Delete everything under this level? [yes,no] (no) yes

[edit shared sae configuration aggregate-services]
user@host# show

[edit shared sae configuration aggregate-services]
user@host#
```

Remove the configuration for a specific property (routing-options):

```
[edit]
user@host# set routing-options static route 192.0.2.20/24 reject

[edit]
user@host# show
. . .
static {
  route 192.0.2.20/24 {
    reject;
  }
}
. . .
[edit]
user@host# delete routing-options

[edit]
user@host# edit routing-options

[edit routing-options]
user@host# show
```

Renaming an Identifier

To modify a configuration, you can rename an identifier that already exists. 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 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 ntp server 10.0.0.7 to server 10.0.0.6
```

Inserting a New Identifier

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, in a few cases the ordering of the statements matters because the configuration statements create a sequence that is analyzed in order.

For example, rules for interface, subscriber, and DHCP classification scripts are evaluated in the order in which they appear in the configuration. If you add a rule that you want to be evaluated before an existing rule, you need to modify the ordering of the rules,.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.

You use the insert command to reorder identifiers that you have already configured.

Examples: Inserting a New Identifier

Add a new subscriber classification rule and insert it before existing rules at the [edit shared sae user-classifier] hierarchy level:

```
[edit shared sae user-classifier]
user@host> show
rule rule-2 {
  target <-retailerDn->??sub?(uniqueID=<-userName->);
  condition {
    loginType == "SYNC";
  }
}
rule rule-3 {
  target <-unauthenticatedUserDn->;
  condition {
    loginType == "TOKEN";
    loginType == "PUBLIC";
  }
}
rule rule-4 {
  target <-retailerDn->??sub?(uniqueID=<-userName->);
  condition {
    retailerDn != "";
    & userName != "";
  }
}

[edit shared sae user-classifier]
user@host# set rule new target "[<-unauthenticatedUserDn->]"

[edit shared sae user-classifier]
user@host# set rule new condition "loginType=="AuthADDR""

[edit shared sae user-classifier]
user@host# insert rule new before rule-2

[edit shared sae user-classifier]
user@host# show
rule new {
  target "[<-unauthenticatedUserDn->]";
  condition {
    loginType==AuthADDR;
  }
}
```

```
rule rule-2 {
  target <-retailerDn->??sub?(uniqueID=<-userName->);
  condition {
    loginType == "SYNC";
  }
}
rule rule-3 {
  target <-unauthenticatedUserDn->;
  condition {
    loginType == "TOKEN";
    loginType == "PUBLIC";
  }
}
rule rule-4 {
  target <-retailerDn->??sub?(uniqueID=<-userName->);
  condition {
    retailerDn != "";
    & userName != "";
  }
}
```

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 there is an error in the configuration syntax, the commit check command returns a message that indicates the location of the error. For example:

```
[edit]
user@host# commit check
[edit shared sae configuration driver]
  missing mandatory attribute(s): 'unauthenticated-subscriber-dn',
'mac-cache-expiration'
```

Committing a Configuration

To save software configuration changes to the directory and activate the configuration:

- Use the commit configuration mode command.

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

When you enter the commit command, the software reviews the configuration for syntax errors (commit check). Then, if the syntax is correct, the configuration is activated and becomes the active configuration.

You can issue the commit command from any hierarchy level.

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:

```
user@host# commit
[edit system login user Chris Bee class]
Undefined class 'newClass'
```

We recommend that you correct the error before recommitting the configuration. To return quickly to the hierarchy level where the error is located, copy the path from the last line of the message 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, 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>
```

When Multiple Users Configure the Software

A number of users can be working 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.

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>
```

Chapter 7

Filtering Command Output in the SRC CLI

For commands that display output, such as the **show** commands, you can filter the output. Topics include:

- Using Keyboard Sequences at the MORE Prompt in the SRC CLI on page 69
- Using the Pipe (|) Symbol When Entering Commands on page 70

Using Keyboard Sequences at the MORE Prompt in the SRC CLI

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.

The SRC software uses the Less program to provide navigation and search capability at the MORE prompt. The SRC software does not permit the Less program access to files or the shell.

Table 11 lists the keyboard sequences for the commands most frequently used at the MORE prompt.

Table 11: MORE Prompt Keyboard Sequences

Category	Action	Keyboard Sequence
Get Help	Display information about the keyboard sequences you can display at the MORE prompt. Commands listed that access files or the shell are not supported.	h, H
Scroll Down	Scroll down one line.	e, Ctrl+e, j, Ctrl+n, Enter, down arrow
	Scroll down one-half screen.	d, Ctrl+d
	Scroll down one whole screen.	f, Ctrl+f, Ctrl+v, z, Space
	Scroll down to the bottom of the output.	F
	Jump to the last line in the output and exit to the CLI prompt.	G

Table 11: MORE Prompt Keyboard Sequences (continued)

Category	Action	Keyboard Sequence
Scroll Up	Display the previous line of output.	y, Ctrl+y, k, Ctrl+k, Ctrl+p, up arrow
	Scroll up one-half screen.	u, Ctrl+u
	Scroll up one whole screen.	b, Ctrl+b, Esc+v, w
	Jump to the first line of the output.	g
Move left and right	Move right one-half screen width.	Esc +), right arrow
	Move left one-half screen width.	ESC + (, left arrow
Search	Search forward for a string. You can also search for a string by specifying the | match filter when entering a command using the pipe symbol. See *Displaying Output That Matches a Regular Expression on page 74*.	/ *string*
	Search backward for a string.	?*string*
	Repeat a previous search for a string.	Up arrow while in search mode
	Repeat the previous search for a string in the opposite direction.	Down arrow while in search mode
Interrupt or End Output, Redraw the Output, and Save the Output to a File	Interrupt the display of output.	q, :q, Q, :Q, ZZ
	Redraw the output on the screen.	r, Ctrl+r, Ctrl+l
	Redraw the output on the screen and discard buffered input.	R

Using the Pipe (|) Symbol When Entering Commands

You can filter output by adding the | (*pipe*) symbol when you enter a command. For example, you can use the pipe symbol after show configuration:

```
user@host> show configuration ?
Possible completions:
  <[Enter]>            Execute this command
> interfaces           Interfaces on the SDX platform
> policies             Policy configuration
> redirect-server      Redirect server properties
> routing-options      Protocol-independent routing option configuration
> services             Service configuration
> shared               Shared configuration
> slot                 Component configuration
> snmp                 SNMP agent
> subscribers          Subscriber and subscription configuration
> system               System parameters
|                    Pipe through a command
```

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

```
user@host> show configuration | ?
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 the pattern
  last                  Display end of output only
  match                 Show only text that does match a pattern
  no-more               Do not paginate output
  save                  Save output text to file
```

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 72* for a description of each type of filter.

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 12.) If the regular expression contains spaces, operators, or wildcard characters, enclose the expression in quotation marks.

Table 12: 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
```

A pipe filter of **|** except 1 displays the following output:

```
2 2
4
```

Related Topics

- *Disregarding Output That Does Not Match a Regular Expression on page 73*
- *Displaying Output from the First Match of a Regular Expression on page 73*
- *Displaying Output That Matches a Regular Expression on page 74*

Pipe Filter Functions

You can perform the following tasks by using the pipe filter:

- Counting the Number of Lines of Output on page 72
- Displaying Output in XML Tag Format on page 72
- Disregarding Output That Does Not Match a Regular Expression on page 73
- Displaying Output from the First Match of a Regular Expression on page 73
- Displaying the End of the Output for a Command on page 74
- Displaying Output That Matches a Regular Expression on page 74
- Preventing Output from Being Paginated on page 75
- Saving Output to a File on page 75

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: 369 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
<?xml version="1.0"?>
<output>
Current directory: /root
</output>
```

Disregarding Output That Does Not Match a Regular Expression

If the regular expression contains any spaces, operators, or wildcard characters, enclose it in quotation marks.

For information about common regular expression operators, see Table 12 on page 71.

To disregard text that matches a regular expression:

- Specify the **except** command after the pipe symbol.

The following example displays information about configuration interfaces with the exception of the family information for each interface:

```
user@host> show configuration interfaces |except family
lo {
  unit 0 {
      inet {
        address 192.0.0.1;
      }
    }
  }
}
eth0 {
  unit 0 {
      inet {
        address 10.27.7.45/24;
      }
    }
  }
}
```

Displaying Output from the First Match of a Regular Expression

When you use regular expressions, enclose any spaces, operators, or wildcard characters in quotation marks.

For information about common regular expression operators, see Table 12 on page 71.

To display output starting with the first occurrence of text matching a regular expression:

- Enter **find** after the pipe symbol.

The following example starts displaying information for the **show system information** command starting with the **Current time** section of the output:

```
user@host> show system information |find time
Current time              2006-10-31 09:34:17 EST
Uptime                    11 days, 17:26
Number of active users    2
Load Averages (1m/5m/15m) 0.09/0.08/0.09

Memory
```

```
Total 15G
Free  14G

CPU Info
Number of CPU 4
CPU Model     Dual Core AMD Opteron(tm) Processor 265
Clock Speed   1804.137 MHz

Disk Information
  Mountpoint       Total Used  Use%
  /                2015M 1018M  50%
  /altroot         2015M 1015M  50%
  /boot              98M   14M  14%
  /var             5039M  497M   9%

. . .
```

Displaying the End of the Output for a Command

To display the end of the output for a command:

- Enter last after the pipe symbol.

The following example shows the end of the output for the show system information command.

```
user@host> show system information |last
Temperature
System +26 C
CPU-1  +39 C
CPU-2  +39 C

Fan Speed
Fan-1 9375 RPM
Fan-2 9375 RPM
```

Displaying Output That Matches a Regular Expression

If the regular expression contains any spaces, operators, or wildcard characters, enclose it in quotation marks.

For information about common regular expression operators, see Table 12 on page 71.

To display output that matches a regular expression:

- Enter match *regular-expression* after the pipe symbol.

The following example matches all Ethernet interfaces in the interface configuration:

```
user@host> show configuration interfaces | match eth
eth0 {
```

Preventing Output from Being Paginated

By default, if output is longer than the length of the terminal screen, a MORE message lets you display the remaining output when you press the Spacebar. You can use the | no-more filter to display all output at once. This feature is useful when you want access to the entire output, such as to copy the entire output and paste it into an e-mail message.

To prevent the output from being paginated:

- Enter no-more after the pipe symbol.

  For example, to display all output from the show configuration command at once:

  user@host> **show configuration | no-more**

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.

NOTE: When you run a show command, your SRC privileges determine the information that you view. Therefore, when you save this information, you are saving only the configuration information that you have privileges to view.

By default, the file is placed in the current working directory of the CLI.

To save command output to a file:

- Enter save *filename* after the pipe symbol.

The following example saves the output from the show command to a file named my-config-info.txt:

```
[edit system]
user@host> show | save my-config-info.txt
Wrote 78 lines of output to ‘my-config-info.txt’
user@host>
```

Chapter 8

Using Keyboard Shortcuts in the SRC CLI

This chapter provides information about how to use keyboard shortcuts to save time when you enter commands and configuration statements.

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 13 lists some of the CLI keyboard sequences. They are the same as those used in Emacs.

Table 13: CLI Keyboard Sequences

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.	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, Alt+Delete, 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

Table 13: CLI Keyboard Sequences (continued)

Category	Action	Keyboard Sequence
Redraw the Screen	Redraw the current line.	Ctrl+l
Display Previous Command Lines	Scroll backward through the list of recently executed commands.	Ctrl+p or up arrow
	Scroll forward through the list of recently executed commands.	Ctrl+n or down arrow
	Search the CLI history in reverse order for lines matching the search string.	Ctrl+r
	Terminate a search, and display the found command.	Esc or Ctrl+j
	Abort a search.	Ctrl+g
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*

Part 3

SRC CLI Environment and SRC Component Management

Chapter 9

Controlling the SRC CLI Environment

This chapter describes how to set up your CLI environment. Topics include:

Overview of Commands to Control the SRC CLI Environment

In operational mode, you can use commands to control the command-line interface (CLI) environment. For example, you can specify the remote terminal type. The following output lists the options that you can use to control the CLI environment settings.

```
user@host> set cli ?
Possible completions:
  complete-on-space    Command completion on partial command entry
  directory            Working directory on the system
  language             Terminal language and encoding
  level                Access level for CLI commands
  password             Change the current password
  prompt               CLI command prompt
  screen-length        Number of lines on the screen
  screen-width         Screen width in columns
  terminal             Terminal type
```

NOTE: When you use SSH to log in to a C-series Controller or log in from the console when its terminal type is already configured, your terminal type, screen length, and screen width are already set.

Setting the Editing Level for the SRC CLI

The editing level determines which statements and commands are visible to a user from the SRC CLI. Table 14 describes the editing levels.

Table 14: Editing Levels

Level	Description
Basic	Only values that must be configured are visible.
Normal	Common values and basic values are visible; this is the default setting.
Advanced	All configurable values, including the common and basic values, are visible.
Expert	All configurable values and internal values used for debugging are visible.

If you log in to the CLI as `root`, the default editing level, normal, is available to you because `root` does not require a user profile to access the CLI. Although `root` access is used for initial configuration of a C-series Controller, user accounts are used to enter commands and statements at the CLI.

The editing level can be set for:

- Specified users in the user profiles
- A current user session.

For information about the editing level of configuration statements and options, see the *SRC-PE CLI Command Reference*.

To set the editing level for the CLI:

- In operational mode, use the set cli level command.
  - set cli level basic
  - set cli level normal
  - set cli level advanced
  - set cli level expert

To view the editing level of the current CLI session:

- In operational mode, use the show cli command.

Setting the Terminal Type for the SRC CLI

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, xterm, or dumb.

Setting the Language for the Terminal Environment for the SRC CLI

To set the language appropriate to the terminal environment:

- In operational mode, use the set cli language *language* command.

  For *language*, specify the language and encoding in the following format:

  2-character language code (lower case)_2-character country code (upper case)

  For example, en_US.UTF8 (the default).

Setting the Screen Length for the SRC CLI

Typically, the terminal used to access the C-series Controller controls the length of the screen. Although you can change the value for the screen length, if the terminal supports reporting the screen size, the screen size reported by the terminal takes precedence.

To change the length of the CLI screen:

- Use the set cli screen-length command.

  ```
  user@host> set cli screen-length length
  ```

Setting the Screen Width for the SRC CLI

Typically, the terminal used to access the C-series Controller controls the width of the screen. Although you can change the value for the screen width, if the terminal supports reporting the screen size, the screen size reported by the terminal takes precedence.

To change the width:

- Use the set cli screen-width command.

  ```
  user@host> set cli screen-width width
  ```

Changing the Password for the SRC CLI

To change the current password used to access the CLI:

- In operational mode, use the set cli password command.

  ```
  user@host> set cli password
  ```

NOTE: If you are using a C-series Controller, the `root` password should have been changed from the default setting when the system was initially configured.

Setting the SRC CLI Prompt

The default CLI prompt is user@host> in operational mode and user@host# in configuration mode.

To change the prompt:

- Use the set cli prompt command.

  ```
  user@host> set cli prompt string
  ```

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

  Specify the characters \> to have > appear at the end of the prompt in operational mode and # at the end of the prompt in configuration mode.

Setting the Directory for the SRC CLI

By default, on a C-series Controller the working directory is the home directory of the user. On a Solaris platform, it is the directory from which you executed the cli command to start the CLI. You can change your working directory for the CLI.

To change the current working directory,:

- Use the set cli directory command.

  user@host> **set cli directory** *directory*

  where *directory* is the pathname of working directory.

Setting Command Completion for the SRC CLI

You can enter a Tab after a partially typed command to complete the command. In addition, you can enable and disable the CLI to complete a command by typing a space for:

- Specified users in the user profiles
- A current user session

To disable command completion when entering a space:

- Use the set cli complete-on-space off command.

  user@host> **set cli complete-on-space off**

To re-enable command completion when entering a space:

- Use the set cli complete-on-space on command.

  user@host> **set cli complete-on-space on**

Viewing Settings for the SRC CLI

To view the current CLI settings:

```
user@host> show cli
CLI complete-on-space set to on
CLI editing level is: normal
CLI working directory is '/'
```

You can also use the show cli directory command to display the current working directory.

To view the authorization settings for the user logged in to the CLI:

```
user@host> show cli authorization
Current user: 'root' class 'super-user'
Permissions:
    admin                    -- Can view user accounts
    admin-control            -- Can modify user accounts
    clear                    -- Can clear learned network information
    configure                -- Can enter configuration mode
    field                    -- Special for field (debug) support
    firewall                 -- Can view firewall configuration
    firewall-control         -- Can modify firewall configuration
    interface                -- Can view interface configuration
    interface-control        -- Can modify interface configuration
    maintenance              -- Can perform system maintenance (as wheel)
    network                  -- Can access the network
    reset                    -- Can reset and restart interfaces and
processes
    routing                  -- Can view routing configuration
    routing-control          -- Can modify routing configuration
    secret                   -- Can view secret configuration
    secret-control           -- Can modify secret configuration
    security                 -- Can view security configuration
    security-control         -- Can modify security configuration
    shell                    -- Can start a local shell
    snmp                     -- Can view SNMP configuration
    snmp-control             -- Can modify SNMP configuration
    system                   -- Can view system configuration
    system-control           -- Can modify system configuration
    view                     -- Can view current values and statistics
    service                  -- Can view service definitions
    service-control          -- Can modify service definitions
    subscriber               -- Can view subscriber profiles
    subscriber-control       -- Can modify subscriber profiles
    Individual command authorization:
    Allow regular expression: none
    Deny regular expression: none
    Allow configuration regular expression: none
    Deny configuration regular expression: none
```

Chapter 10

Managing SRC Configurations

This chapter provides basic information about managing configurations. Topics include:

How the SRC Configuration Is Stored

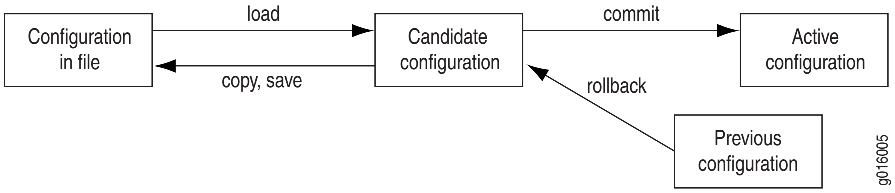
When you edit a configuration, you work in a copy of the current configuration to create a candidate configuration. Changes you make to the candidate configuration are visible in the CLI immediately. If multiple users edit the configuration at the same time, all users can see all changes.

To have a candidate configuration take effect, you *commit* the changes. At this point, the software verifies the candidate configuration for proper syntax. If multiple users are editing the configuration, when you commit the candidate configuration, all changes made by all the users take effect.

Slot (local) configuration is stored in files, and the remainder of the configuration is stored in the Juniper Networks Database or another directory that you have configured to store SRC configuration data.

Figure 10 illustrates the various configuration states and the configuration mode commands that you use to load, commit, copy, and save the configuration.

Figure 10: Commands for Storing and Modifying the Configuration



Updating the SRC Configuration

You can update the SRC configuration to include configuration changes from a file or to revert to the configuration supplied with the SRC product. You can also retain the active configuration and discard changes not yet committed. After you load one of these configurations, you can commit it to activate the configuration on the C-series Controller, or you can edit the configuration interactively using the CLI and commit it at a later time.

Before You Load a Configuration

Before you load a configuration, make a copy of the current configuration. This configuration contains the active configuration plus any configuration changes that have been made at the CLI. For information about the syntax in the files, see:

- Text format—*SRC-PE CLI Command Reference, Volume 1* and *SRC-PE CLI Command Reference, Volume 2*
- XML format—*SRC-PE XML API Configuration Reference*

You can save the configuration to text or XML format. By default, the configuration is saved to a file in XML format.

To make a backup copy of the configuration:

- From the [edit] hierarchy level of configuration mode, save the configuration to XML format. For example:

```
[edit]
user@host> save backupcfg.xml format xml
```

- From the [edit] hierarchy level of configuration mode, save the configuration to text format. For example:

```
[edit]
user@host> save backupcfg.txt format text
```

Commands to Load a Configuration

You can use the following commands in configuration mode to make configuration changes:

- load factory-default—Replace the existing configuration with the configuration supplied with the SRC software.
- load merge *filename* <relative>—Combine the configuration that is currently shown in the CLI and the configuration in the specified file.
- load override *filename* —Discard the entire configuration that is currently shown in the CLI, and load the entire configuration in the specified file.
- load replace *filename* <relative>— Look for replace attributes in the specified file, delete the existing statements of the same name, and replace them with the configuration in the specified file.
- load set *filename* <relative>—Execute configuration mode commands such as set, edit, exit, and top from a text file.

The relative option for the load merge, load replace, and load set commands lets you load the configuration at a specified hierarchy level.

The load merge, load override, and load replace commands let you update configuration statements in the SRC configuration from a text file or an XML file. The structure of this file must conform to the structure for an SRC configuration file. For this reason, we recommend that you copy the file based on the file format you plan to use:

- XML format—Save a configuration to an XML file and modify that file.
- Text format—Save a configuration to a text file, or copy the output from a show command to a file, and modify that file.

For a merge or replace operation, you can save a copy of the configuration at any level in the configuration hierarchy, then load the updated configuration at the same level.

Use the editor of your choice to modify a saved configuration file. When you edit a file that is to be loaded into the SRC configuration, you can add specified attributes to specify actions to be taken.

Attributes in SRC Configuration Files

You can add the following attributes to text files or to XML tags in a configuration file to be loaded through the load merge, load override, and load replace commands. If you do not add any attributes, the software merges all changes.

- operation=”create”—Create the specified configuration.

  If you try to create a configuration object that already exists, the software does not create the new objects and generates an error message to that effect.

- operation=”delete”—Delete the specified configuration.

- operation=”merge”—Merge the specified configuration.

- operation=”replace”—Replace a specified configuration with another defined configuration.

  If the replace attribute is in the file whose contents are merged, the command disregards the replace attribute.

About SRC Configuration Files in XML Format

The XML structure follows the same hierarchy as the CLI. For example, in configuration mode the following statements are available at the [edit system] hierarchy level:

```
[edit system]
user@host# set ?
Possible completions:
+ authentication-order Order in which authentication methods are invoked
+ domain-search        List of domain names to search
  host-name            Hostname for C-series Controller
> ldap                 LDAP properties
> login                Login properties
+ name-server          DNS name servers
> ntp                  Configure NTP
> radius-server        RADIUS server configuration
> services             System services configuration
> syslog               System log configuration
> tacplus-server       TACACS+ server configuration
  time-zone            Time zone definition name
```

In an XML file, the tags within the <system> tags are the same as the statements in the [edit system] hierarchy. The tags under <system> can appear in any order.

```
<configuration>
    <system>
      <authentication-order> </authentication-order>
      <domain-search> </domain-search>
      <host-name> </host-name>
      <ldap> </ldap>
      <login> </login>
      <name-server> </name-server>
      <ntp> </ntp>
      <radius-server> </radius-server>
      <services> </services>
      <syslog> </syslog>
      <tacplus-server> </tacplus-server>
      <time-zone> </time-zone>
    </system>
</configuration>
```

The following example shows parts of a configuration file for statements in the [edit system] hierarchy:

```
<?xml version="1.0"?>
<configuration>
    <system>
      <time-zone>Canada/Eastern</time-zone>
      <services>
        <telnet/>
        <ssh>
          <root-login>allow</root-login>
        </ssh>
      </services>
      <host-name>myhost</host-name>
      <name-server>192.2.2.10</name-server>
      <name-server>192.2.2.20</name-server>
      <domain-search>mydomain.juniper.net</domain-search>
      <domain-search>juniper.net</domain-search>
      <ntp>
        <server> <address>192.2.2.100</address>
        </server>
        <boot-server>192.2.2.100</boot-server>
      </ntp>
      <ldap>
        <server>
          <address>10.227.2.100</address>
        </server>
        <boot-server>10.227.2.100</boot-server>
      </ldap>
      <ldap>
        <server>
          <community>
            <primary-neighbors>neighbor1</primary-neighbors>
            <role>primary</role>
          </community>
        </server>
      </ldap>
      <ldap>
        <client>
          <connection-manager-id>CLI_DATA_MANAGER
          </connection-manager-id>
        . . .
        </client>
      </ldap>
      <login>
        <class>
          <name>class-cfg</name>
          <allow-configuration>s.*m$|s.*m l.*n</allow-configuration>
          <permissions>configure</permissions>
          <permissions>interface</permissions>
        </class>
        <user>
          <user-name>admin</user-name>
          <class>super-user</class>
          <full-name>admin</full-name>
          <uid>500</uid>
          <gid>100</gid>
          <authentication>
            . . .
          </authentication>
          <level>normal</level>
          <complete-on-space>on</complete-on-space>
        </user>
      </login>
      <syslog>
        . . .
      </syslog>
    </system>
</configuration>
```

Example: Using Attributes When Editing an XML Configuration File

You can modify a single value by inserting an attribute into one tag. For example, to delete the name server that has the IP address 192.2.2.20:

```
<configuration>
    <system>
      <name-server operation=”delete”>192.2.2.20</name-server>
    </system>
</configuration>
```

You can also modify a number of values within a hierarchy by adding an attribute at a higher level in the hierarchy. For example, to replace permissions for the class named class-cfg in the following configuration:

```
<configuration>
    <system>
        <class>
          <name>class-cfg</name>
          <allow-configuration>s.*m$|s.*m I.*n</allow-configuration>
          <permissions>configure</permissions>
          <permissions>interface</permissions>
        </class>
    </system>
</configuration>
```

Enter the replace attribute for the class:

```
<configuration>
    <system>
      <login>
        <class operation=”replace”>
          <name>class-cfg</name>
          <allow-configuration>s.*m$|s.*m l.*n</allow-configuration>
          <permissions>control</permissions>
          <permissions>maintenance</permissions>
        </class>
      </login>
    </system>
</configuration>
```

About SRC Configuration Files in Text Format

You can create a configuration file in text format by saving the configuration to a file in text format or by running the show command at a specified hierarchy level, and then copying the output into a text file. The hierarchical format you see when you run a show command shows the statement hierarchy as it appears in a text file.

You can also create a text file that includes configuration mode commands to be executed and then load this file though the load set command. Use the editor of your choice to create the text file.

For example, to add a name server that has the IP address 192.2.2.30 and to delete the name server that has the IP address 192.2.2.20 add the following lines to a text file:

```
edit system
set name-server 192.2.2.30
delete name-server 192.2.2.20
```

Example: Using Attributes When Editing a Text Configuration File

You can modify a single value by inserting an attribute. For example, to delete the name server that has the IP address 192.2.2.20:

```
configuration{
    system{
      delete:
      name-server 192.2.2.20;
    {
{
```

You can also modify a number of values within a hierarchy by adding an attribute at a higher level in the hierarchy. For example, to replace permissions for the class named class-cfg in the following configuration:

```
configuration{
    system{
        class{
          name class-cfg;
          allow-configuration s.*m$|s.*m l.*n;
          permissions configure;
          permissions interface;
        }
    }
}
```

Enter the replace attribute for the class:

```
configuration{
    system{
      login{
        replace:
        class{
          name class-cfg;
          allow-configuration s.*m$|s.*m l.*n;
          permissions control;
          permissions maintenance;
        }
      }
    }
}
```

Preparing a File to Be Loaded into the Current SRC Configuration

When you save your current configuration to a file, the file contains 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.

When you save a configuration to a file, the contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it. When you save a file to XML format, the software inserts a line in the saved file to indicate the level at which the file was saved. For example:

```
<configuration>
    <system>
      <services sdx:current="true">
        <ssh>
          <root-login>deny</root-login>
        </ssh>
        <editor>
          <password-encryption>sha</password-encryption>
        </editor>
      </services>
    </system>
</configuration>
```

The file is saved in the current working directory. When you load a file that was saved at a specific hierarchy level, use the relative option for a load command. If you do not use the relative option, the command disregards the sdx:current="true" text in XML files.

If you plan to copy a configuration file from the C-series Controller to another system and back, make sure that you have SSH or Telnet enabled on the C-series Controller.

The examples in the following procedure show how to prepare a file in XML format; the procedure is the same for files in text format.

To prepare a configuration file for loading into the SRC configuration:

1. In configuration mode, navigate to the level at or below which you want to save the configuration.
2. Run the save command.

   For example:

   ```
   [edit system]
   user@host# save systemcfg.xml
   172 lines written to systemcfg.xml
   [edit system]
   ```

3. Edit the file.

   On a C-series Controller:

   a. Copy the file to a remote system, and then edit it. For example:

```
user@host> file copy /root/systemcfg.xml ftp://user@myserver/systemcfg.xml
Password:

user@host>
```

   For information about specifying the filename, see *Chapter 5, Using the SRC CLI Operational Commands to Monitor the SRC Software*.

   b. Edit the file in the editor of your choice.

c. Copy the edited file back to the C-series Controller. For example:

```
user@host> file copy ftp://user@myserver/systemcfg.xml /root/systemcfg2.xml
Password:

user@host>
```

On a Solaris platform:

- Edit the file in the text editor of your choice.

Related Topics

- For information about enabling SSH and Telnet on the C-series Controller, see *SRC-PE Getting Started Guide, Chapter 10, Managing SRC Configurations*.

Loading an SRC Configuration

You can use the load commands to perform the following tasks:

- *Replacing the Current Configuration with the Default SRC Configuration on page 96*
- *Merging the Active Configuration with Another Configuration on page 96*
- *Replacing the Configuration on page 98*
- *Replacing Parts of the Configuration on page 99*
- *Adding a Configuration Through Configuration Mode Commands on page 101*
- *Loading a Configuration at a Specified Hierarchy Level on page 101*

Replacing the Current Configuration with the Default SRC Configuration

To restore the full default SRC configuration:

- In configuration mode, enter the load factory-default command.

  ```
  [edit]
  user@host# load factory-default
  ```

  This command removes the active configuration and replaces it with the basic, default SRC configuration.

Merging the Active Configuration with Another Configuration

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.

You can merge a configuration from files in XML or text format. The examples in this section use files in XML format.

You can merge all of the configuration, or the configuration at a specified hierarchy level. For information about loading a configuration at a specified hierarchy level, see *Loading a Configuration at a Specified Hierarchy Level on page 101*.

To combine the active configuration and the configuration in a specified file:

- In configuration mode, specify the load merge command. For example:

```
[edit]
user@host# load merge newcfg.xml format xml
```

The following example shows part of an existing configuration, the configuration in the file to be loaded, and the resulting configuration. In the resulting configuration, bold text indicates the configuration that changed.

Existing configuration:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.10</name-server>
      <name-server>192.2.2.20</name-server>
      <domain-search>mydomain.juniper.net</domain-search>
      <domain-search>juniper.net</domain -search>
      . . .
    </system>
    . . .
</configuration>
```

Configuration in the file to be loaded:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.30</name-server>
      <domain-search>newdomain.juniper.net
      </domain-search>
      . . .
      </system>
    . . .
</configuration>
```

Resulting configuration:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.10</name-server>
      <name-server>192.2.2.20</name-server>
      <name-server>192.2.2.30</name-server>
```

```
            <domain-search>mydomain.juniper.net</domain-search>
            <domain-search>juniper.net</domain-search>
            <domain-search>newdomain.juniper.net</domain-search>
            . . .
        </system>
        . . .
</configuration>
```

Replacing the Configuration

You can replace a configuration from files in XML or text format. The examples in this section use files in XML format.

To replace all of the active configuration with a full configuration in a specified file:

- In configuration mode, specify the load override command. For example:

  ```
  [edit]
  user@host# load override complete-newcfg.xml format xml
  ```

  When you use the load override command and commit the configuration, all system processes reparse the configuration.

The following example shows part of an existing configuration, the configuration in the file to be loaded, and the resulting configuration. In the resulting configuration, bold text indicates the configuration that changed.

Existing configuration:

```
<configuration>
    . . .
    <system>
        . . .
        <host-name>myhost</host-name>
        <name-server>192.2.2.10</name-server>
        <name-server>192.2.2.20</name-server>
        <domain-search>mydomain.juniper.net</domain-search>
        <domain-search>juniper.net</domain -search>
        . . .
</system>
    . . .
</configuration>
```

Configuration in the file to be loaded:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.30</name-server>
      <name-server>192.2.2.40</name-server>
      <domain-search>newdomain.juniper.net
      </domain-search>
      . . .
    </system>
    . . .
</configuration>
```

Resulting configuration:

```
<configuration>
    . . .
    <system>
    . . .
    <host-name>myhost</host-name>
      <name-server>192.2.2.30</name-server>
      <name-server>192.2.2.40</name-server>
      <domain-search>newdomain.juniper.net</domain-search>
    . . .
      </system>
    . . .
</configuration>
```

Replacing Parts of the Configuration

A replace operation searches for **replace** attributes in the specified file, 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** attribute. You can also use **create**, **delete**, and **merge** attributes in the file.

If you are performing a replace operation and the file you specify does not contain any **replace** attributes, the replace operation is effectively equivalent to a merge operation. This type of operation 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.

You can merge a configuration from files in XML or text format. The examples in this section use files in XML format.

You can replace all of the configuration, or the configuration at a specified hierarchy level. For information about loading a configuration at a specified hierarchy level, see *Loading a Configuration at a Specified Hierarchy Level on page 101*.

To replace portions of a configuration:

1. Make sure that the incoming configuration file has replace attributes in place for each part of the configuration to be replaced.

   See *Example: Using Attributes When Editing an XML Configuration File on page 92*.

2. In configuration mode, specify the load replace command. For example:

   ```
   user@host# load replace newcfg.xml format xml
   ```

The following example shows part of the existing configuration, the configuration in the file to be loaded, and the resulting configuration. In the resulting configuration, bold text indicates the configuration that changed.

For an example of a file snippet that shows how to replace a number a values within a hierarchy, see *Example: Using Attributes When Editing an XML Configuration File on page 92*.

Existing configuration:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.10</name-server>
      <name-server>192.2.2.20</name-server>
      <domain-search>mydomain.juniper.net</domain-search>
      <domain-search>juniper.net</domain -search>
      . . .
</system>
    . . .
</configuration>
```

Configuration in the file to be loaded:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      < name-server operation=”replace”>192.2.2.10</name-server>
      <name-server>192.2.2.30
      </name-server>
      <domain-search>mydomain.juniper.net</domain-search>
      <domain-search>mydomain.juniper.net
      </domain-search>
      <domain-search>juniper.net</domain -search>
      . . .
    </system>
    . . .
</configuration>
```

Resulting configuration:

```
<configuration>
    . . .
    <system>
    . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.10</name-server>
      <name-server>192.2.2.30</name-server>
      <domain-search>mydomain.juniper.net</domain-search>
      <domain-search>juniper.net</domain -search>
      . . .
    </system>
    . . .
</configuration>
```

Adding a Configuration Through Configuration Mode Commands

When you use the load set command, it executes the configuration instructions line by line as they are stored in a file. The instructions can contain any configuration mode command, such as set, edit, exit, and top.

To load a configuration that contains configuration mode commands:

1. Create a text file that includes set and other configuration mode commands. For example:

   ```
   edit system login class name newclass permissions system
   delete system login class name newclass permissions interface
   set system login class name newclass permissions configure
   ```

2. In configuration mode, use the load set command.

   ```
   user@host# load set newcfg2.txt
   ```

Loading a Configuration at a Specified Hierarchy Level

The load merge, load replace, and load set commands let you load the configuration at a specified hierarchy level by using the relative option.

To load a configuration at a hierarchy level:

1. In configuration mode, move to the hierarchy level at which you want to load a configuration.

2. At the hierarchy level, enter a load merge, load replace, or load set command. For example:

   ```
   [edit system login class name newclass]
   user@host# load merge mynewcfg.xml relative format xml
   ```

Reverting to a Previous SRC Configuration

You can revert to the active configuration and discard configuration changes not yet committed.

To revert to the full committed configuration:

- In configuration mode, at the [edit] hierarchy level enter the rollback command.

  ```
  user@host> rollback
  ```

Cutting and Pasting Configuration Information at the SRC CLI

You can also create a configuration by cutting and pasting existing portions of the configuration. You can copy configuration text from another source or from another part of the configuration to a new location. Use the cut and paste functions for your windowing system, such as X Windows.

Chapter 11

Managing SRC Components with the CLI

This chapter describes how you can start, stop, and view status of SRC components from the CLI. Topics include:

- Verifying Status of SRC Components on page 103
- Enabling SRC Components on page 104
- Disabling an SRC Component on page 104
- Restarting an SRC Component on page 105

Verifying Status of SRC Components

To view information about the status for SRC components:

```
user@host> show component
Installed Components
Name      Version                                            Status
cli       Release: 7.0 Build: CLI.A.7.0.0.0171               running
acp       Release: 7.0 Build: ACP.A.7.0.0.0174               disabled
jdb       Release: 7.0 Build: DIRXA.A.7.0.0.0176             running
editor    Release: 7.0 Build: EDITOR.A.7.0.0.0176            disabled
redir     Release: 7.0 Build: REDIR.A.7.0.0.0176             disabled
licSvr    Release: 7.0 Build: LICSVR.A.7.0.0.0179            stopped
nic       Release: 7.0 Build: GATEWAY.A.7.0.0.0170           disabled
sae       Release: 7.0 Build: SAE.A.7.0.0.0166               running
www       Release: 7.0 Build: UMC.A.7.0.0.0169               disabled
jps       Release: 7.0 Build: JPS.A.7.0.0.0172               disabled
agent     Release: 7.0 Build: SYSMAN.A.7.0.0.0174            disabled
webadm    Release: 7.0 Build: WEBADM.A.7.0.0.0173            disabled
```

Enabling SRC Components

On a C-series Controller, you can enable all SRC components from the CLI. On a Solaris platform, you can enable the subset of components supported by the CLI.

You can enable the following SRC components from the CLI:

- Admission Control Plug-In (ACP)
- Service activation engine (SAE)
- C-Web
- Juniper Networks database
- Juniper Policy Server (JPS)
- Network Information Collector (NIC)
- Policy and Services Editor
- Redirect Server
- SNMP agent

To enable a component:

- In operational mode, use the enable *component* command. For example:

  ```
  user@host> enable sae
  ```

Disabling an SRC Component

On a C-series Controller, you can disable a running SRC component from the CLI. On a Solaris platform, you can disable a running component that is supported by the CLI.

To disable a component:

1. Verify which components are running by entering the show component command in operation mode:

   ```
   user@host> show component
   Installed Components
   Name      Version                                      Status
   cli       Release: 7.0 Build: CLI.A.7.0.0.0171         running
   acp       Release: 7.0 Build: ACP.A.7.0.0.0174         disabled
   jdb       Release: 7.0 Build: DIRXA.A.7.0.0.0176       running
   editor    Release: 7.0 Build: EDITOR.A.7.0.0.0176      disabled
   redir     Release: 7.0 Build: REDIR.A.7.0.0.0176       disabled
   licSvr    Release: 7.0 Build: LICSVR.A.7.0.0.0179      stopped
   nic       Release: 7.0 Build: GATEWAY.A.7.0.0.0170     disabled
   sae       Release: 7.0 Build: SAE.A.7.0.0.0166         running
   www       Release: 7.0 Build: UMC.A.7.0.0.0169         disabled
   jps       Release: 7.0 Build: JPS.A.7.0.0.0172         disabled
   ```

```
agent     Release: 7.0 Build: SYSMAN.A.7.0.0.0174           disabled
webadm    Release: 7.0 Build: WEBADM.A.7.0.0.0173           disabled
```

2. Disable a component by using the **disable component** command in operational mode. For example:

   user@host> **disable sae**

Restarting an SRC Component

If an SRC component is enabled, you can restart it if needed. You can use one of the following methods to restart a component:

- gracefully— Shuts down the component, then starts it again. (Default)
- immediately— Sends a signal kill (SIGKILL) signal to immediately stop the component, then starts it again.
- soft—Sends a signal hangup (SIGHUP) signal to the process for the component, then starts it again.

To restart an SRC component:

- In operational mode, use the **restart component** command.

  user@host **restart component** *component* <gracefully | immediately | soft>

  For example, to restart the SAE gracefully:

  user@host **restart component sae gracefully**

Index