



SRC PE Software

CLI User Guide

Release

4.1.x



Published: 2011-05-11

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

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

SRC PE Software CLI User Guide
Release 4.1.x
Copyright © 2011, Juniper Networks, Inc.
All rights reserved.

Revision History
May 2011—Revision 1

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

YEAR 2000 NOTICE

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

SOFTWARE LICENSE

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

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

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

END USER LICENSE AGREEMENT

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

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) 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, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the 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 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 or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations 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 Embedded Software on non-Juniper equipment; (j) use Embedded 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 payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

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 présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

	About the Documentation	xvii
Part 1	Introduction	
Chapter 1	Introducing the SRC CLI	3
Chapter 2	Getting Started: A Quick Tour of the SRC CLI	7
Chapter 3	SRC CLI Basics	19
Chapter 4	Getting Online Help for the SRC CLI	31
Part 2	Operational Mode and Configuration Mode	
Chapter 5	Using the SRC CLI Operational Commands to Monitor the SRC Software	39
Chapter 6	Using Commands and Statements to Configure the SRC Software	51
Chapter 7	Filtering Command Output in the SRC CLI	71
Chapter 8	Using Keyboard Shortcuts in the SRC CLI	81
Part 3	SRC CLI Environment and SRC Module and Component Management	
Chapter 9	Controlling the SRC CLI Environment	85
Chapter 10	Managing SRC Configurations	93
Chapter 11	Managing SRC Modules and Components with the CLI	113
Part 4	Index	
	Index	119

Table of Contents

	About the Documentation	xvii
	SRC Documentation and Release Notes	xvii
	Audience	xvii
	Documentation Conventions	xvii
	Documentation Feedback	xix
	Requesting Technical Support	xix
	Self-Help Online Tools and Resources	xx
	Opening a Case with JTAC	xx
Part 1	Introduction	
Chapter 1	Introducing the SRC CLI	3
	Overview of the SRC CLI	3
	Understanding SRC CLI Command Modes	3
	Understanding SRC Command and Statement Hierarchies	4
	SRC CLI Command Hierarchy	4
	Configuration Statement Hierarchy	5
	Key Features of the SRC CLI	5
	Leveraging Industry-Standard Technologies	6
Chapter 2	Getting Started: A Quick Tour of the SRC CLI	7
	Before You Start the SRC CLI	7
	Starting the SRC CLI	8
	Displaying Commands	8
	Verifying System Status	9
	Configuring a User Account	11
	Creating a Configuration	13
	Using Shortcuts to Create a Configuration	15
	Making Changes to the Configuration	15
	Rolling Back Configuration Changes	16
Chapter 3	SRC CLI Basics	19
	Elements of the Command-Line Interface	19
	SRC CLI Messages	20
	Displaying Command Output	21
	Types of SRC Commands and Statements	23
	Command Options	24
	Configuration Statements and Identifiers	24

	Privilege Levels for Using Commands and Statements	25
	Switching Between Operational Mode and Configuration Mode	26
	Switching to Configuration Mode	26
	Returning to Operational Mode	26
	Running Operational Mode Commands from Configuration Mode	27
	Moving Among Hierarchy Levels in the SRC CLI	28
	Displaying SRC CLI Command History	29
Chapter 4	Getting Online Help for the SRC CLI	31
	Getting Help for Commands and Statements for the SRC CLI	31
	Getting Help for Omitted Statements	32
	Using SRC CLI Command Completion	33
	Using Command Completion in Configuration Mode	33
Part 2	Operational Mode and Configuration Mode	
Chapter 5	Using the SRC CLI Operational Commands to Monitor the SRC Software	39
	SRC CLI Command Categories	39
	Commonly Used Operational Mode Commands	40
	Viewing Files and Directories	41
	Directories on the C Series Controller	41
	Listing Files and Directories	42
	Specifying Filenames and URLs	44
	Managing SRC Modules and Components	45
	Viewing C Series Controller Information	45
	Restarting an SRC Module or Component	47
	Stopping the SRC Software	47
	Rebooting the SRC Software	49
	Viewing Information about Users Logged Into the SRC Software	49
Chapter 6	Using Commands and Statements to Configure the SRC Software	51
	Understanding SRC CLI Configuration Mode	51
	Configuration Mode Commands	51
	Configuration Statements	53
	Configuration Statement Hierarchy	53
	Working in Configuration Mode with the SRC CLI	55
	Creating an SRC Configuration	55
	Entering Configuration Mode	55
	Exiting from Configuration Mode	56
	Modifying the Configuration	57
	Commands to Modify a Configuration	57
	Entering Values for Statement Options	58
	Displaying the Current Configuration	59
	Adding Configuration Statements and Identifiers	62
	Deleting a Statement from the Configuration	63
	Renaming an Identifier	65
	Inserting a New Identifier	65
	Copying a Configuration from One Configuration Location to Another	67

	Displaying Set Commands for the Configuration (SRC CLI)	67
	Verifying a Configuration	68
	Committing a Configuration	68
	Committing a Configuration and Exiting Configuration Mode	69
	When Multiple Users Configure the Software	70
Chapter 7	Filtering Command Output in the SRC CLI	71
	Using Keyboard Sequences at the MORE Prompt in the SRC CLI	71
	Using the Pipe () Symbol When Entering Commands	73
	Using Regular Expressions with the Pipe Symbol	73
	Using Pipe Filter Functions	74
	Counting the Number of Lines of Output	75
	Displaying Summarized Output for a Hierarchy Level (SRC CLI)	75
	Displaying Output in XML Tag Format	76
	Disregarding Output That Does Not Match a Regular Expression	76
	Displaying Output from the First Match of a Regular Expression	77
	Displaying the End of the Output for a Command	78
	Displaying Output That Matches a Regular Expression	78
	Preventing Output from Being Paginated	79
	Saving Output to a File	79
Chapter 8	Using Keyboard Shortcuts in the SRC CLI	81
Part 3	SRC CLI Environment and SRC Module and Component Management	
Chapter 9	Controlling the SRC CLI Environment	85
	Overview of Commands to Control the SRC CLI Environment	85
	Setting the Editing Level for the SRC CLI	86
	Setting the Terminal Type for the SRC CLI	87
	Setting the Language for the Terminal Environment for the SRC CLI	87
	Setting the Screen Length for the SRC CLI	88
	Setting the Screen Width for the SRC CLI	88
	Changing the Password for the SRC CLI	88
	Setting the SRC CLI Prompt	89
	Setting the Directory for the SRC CLI	89
	Setting Command Completion for the SRC CLI	90
	Viewing Settings for the SRC CLI	90
Chapter 10	Managing SRC Configurations	93
	How the SRC Configuration Is Stored	93
	Commands to Save a Configuration to a File	94
	SSH Host Keys	95
	Component State	95
	Updating the SRC Configuration	95
	SRC Configuration Updates	95
	Before You Load a Configuration	96
	Commands to Load a Configuration	96
	SSH Host Keys	98
	Component State	98
	Attributes in SRC Configuration Files	98

	About SRC Configuration Files in XML Format	99
	Example: Using Attributes When Editing an XML Configuration File	101
	About SRC Configuration Files in Text Format	101
	Example: Using Attributes When Editing a Text Configuration File	102
	Preparing a File to Be Loaded into the Current SRC Configuration	103
	Loading an SRC Configuration	104
	Replacing the Current Configuration with the Default SRC Configuration . .	104
	Merging the Active Configuration with Another Configuration	105
	Replacing the Configuration	106
	Replacing Parts of the Configuration	108
	Adding a Configuration Through Configuration Mode Commands	109
	Loading a Configuration at a Specified Hierarchy Level	110
	Comparing SRC Configurations	110
	Reverting to a Previous SRC Configuration	111
	Cutting and Pasting Configuration Information at the SRC CLI	112
Chapter 11	Managing SRC Modules and Components with the CLI	113
	Verifying Status of SRC Components	113
	Enabling SRC Components	113
	Disabling an SRC Component	114
	Restarting an SRC Component	115
Part 4	Index	
	Index	119

List of Figures

Part 1	Introduction	
Chapter 1	Introducing the SRC CLI	3
	Figure 1: Committing a Configuration	4
	Figure 2: CLI Command Hierarchy Example	4
	Figure 3: Configuration Statement Hierarchy Example	5
Chapter 3	SRC CLI Basics	19
	Figure 4: Elements of the Command-Line Interface	19
	Figure 5: Command Prompt in Configuration Mode	20
	Figure 6: Hierarchy-Level Banner	20
	Figure 7: The MORE Prompt	22
	Figure 8: Command Options	24
Part 2	Operational Mode and Configuration Mode	
Chapter 6	Using Commands and Statements to Configure the SRC Software	51
	Figure 9: Sample Configuration Mode Hierarchy of Statements	54
Part 3	SRC CLI Environment and SRC Module and Component Management	
Chapter 10	Managing SRC Configurations	93
	Figure 10: Commands for Storing and Modifying the Configuration	94

List of Tables

	About the Documentation	xvii
	Table 1: Notice Icons	xviii
	Table 2: Text Conventions	xviii
Part 1	Introduction	
Chapter 3	SRC CLI Basics	19
	Table 3: MORE Prompt Keyboard Sequences	22
	Table 4: CLI Configuration Mode Navigation Commands	28
Part 2	Operational Mode and Configuration Mode	
Chapter 5	Using the SRC CLI Operational Commands to Monitor the SRC Software	39
	Table 5: Commonly Used Operational Mode Commands	40
	Table 6: Directories on a C Series Controller	41
	Table 7: Options to Restart an SRC Module or Component	47
Chapter 6	Using Commands and Statements to Configure the SRC Software	51
	Table 8: Summary of Configuration Mode Commands	51
	Table 9: Configuration Mode Top-Level Statements	53
Chapter 7	Filtering Command Output in the SRC CLI	71
	Table 10: MORE Prompt Keyboard Sequences	71
	Table 11: Common Regular Expression Operators in Operational Mode Commands	74
Chapter 8	Using Keyboard Shortcuts in the SRC CLI	81
	Table 12: CLI Keyboard Sequences	81
Part 3	SRC CLI Environment and SRC Module and Component Management	
Chapter 9	Controlling the SRC CLI Environment	85
	Table 13: Editing Levels	86

About the Documentation

- SRC Documentation and Release Notes on page xvii
- Audience on page xvii
- Documentation Conventions on page xvii
- Documentation Feedback on page xix
- Requesting Technical Support on page xix

SRC Documentation and Release Notes

For a list of related SRC documentation, see <http://www.juniper.net/techpubs/>.

If the information in the latest *SRC Release Notes* differs from the information in the SRC guides, follow the *SRC Release Notes*.

Audience

This documentation is intended for experienced system and network specialists working with routers running Junos OS and JunosE software in an Internet access environment. We assume that readers know how to use the routers, directories, and RADIUS servers that they will deploy in their SRC networks. 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

Table 1 on page xviii defines the notice icons used in this guide. Table 2 on page xviii defines text conventions used throughout this 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 or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2: Text Conventions

Convention	Description	Examples
Bold text like this	<ul style="list-style-type: none"> Represents keywords, scripts, and tools in text. Represents a GUI element that the user selects, clicks, checks, or clears. 	<ul style="list-style-type: none"> Specify the keyword exp-msg. Run the install.sh script. Use the pkgadd tool. To cancel the configuration, click Cancel.
Bold text like this	Represents text that the user must type.	user@host# set cache-entry-age cache-entry-age
Fixed-width text like this	Represents information as displayed on your terminal's screen, such as CLI commands in output displays.	<pre>nic-locators { login { resolution { resolver-name /realms/ login/A1; key-type LoginName; value-type SaeId; } } }</pre>
Regular sans serif typeface	<ul style="list-style-type: none"> Represents configuration statements. Indicates SRC CLI commands and options in text. Represents examples in procedures. Represents URLs. 	<ul style="list-style-type: none"> 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/src/api-index.html
<i>Italic sans serif typeface</i>	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.

Table 2: Text Conventions (*continued*)

Key names linked with a plus sign (+)	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
<i>Italic typeface</i>	<ul style="list-style-type: none"> Emphasizes words. Identifies book names. Identifies distinguished names. Identifies files, directories, and paths in text but not in command examples. 	<ul style="list-style-type: none"> There are two levels of access: <i>user</i> and <i>privileged</i>. <i>SRC PE Getting Started Guide</i> <i>o=Users, o=UMC</i> The <i>/etc/default.properties</i> file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	Plugin.radiusAcct-1.class=\net.juniper.smgmt.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

Documentation Feedback

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

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://www.juniper.net/alerts/>
- Join and participate in the Juniper Networks Community Forum: <http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/> .
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html> .

PART 1

Introduction

- Introducing the SRC CLI on page 3
- Getting Started: A Quick Tour of the SRC CLI on page 7
- SRC CLI Basics on page 19
- Getting Online Help for the SRC CLI on page 31

CHAPTER 1

Introducing the SRC CLI

- Overview of the SRC CLI on page 3
- Understanding SRC CLI Command Modes on page 3
- Understanding SRC Command and Statement Hierarchies on page 4
- Key Features of the SRC CLI on page 5

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. 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 `comm` and `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.

Related Documentation

- Elements of the Command-Line Interface on page 19
- Overview of Configuration for the SRC CLI
- Types of SRC Commands and Statements on page 23
- Understanding SRC CLI Command Modes on page 3
- Understanding SRC Command and Statement Hierarchies on page 4

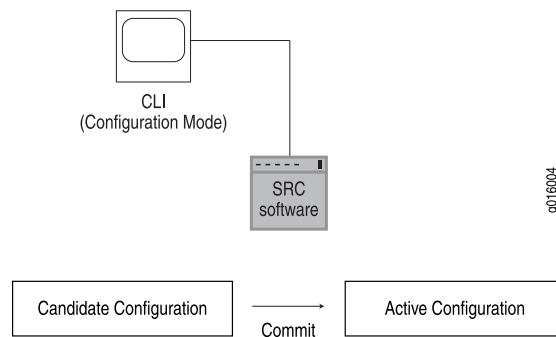
Understanding SRC CLI Command Modes

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



Related Documentation

- SRC CLI Command Categories on page 39
- Commonly Used Operational Mode Commands on page 40
- Understanding SRC CLI Configuration Mode on page 51
- Understanding SRC Command and Statement Hierarchies on page 4
- Key Features of the SRC CLI on page 5

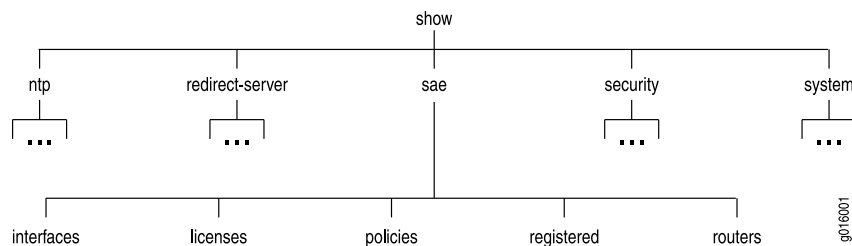
Understanding SRC Command and Statement Hierarchies

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 on page 4 illustrates a portion of the **show** command hierarchy.

Figure 2: CLI Command Hierarchy Example



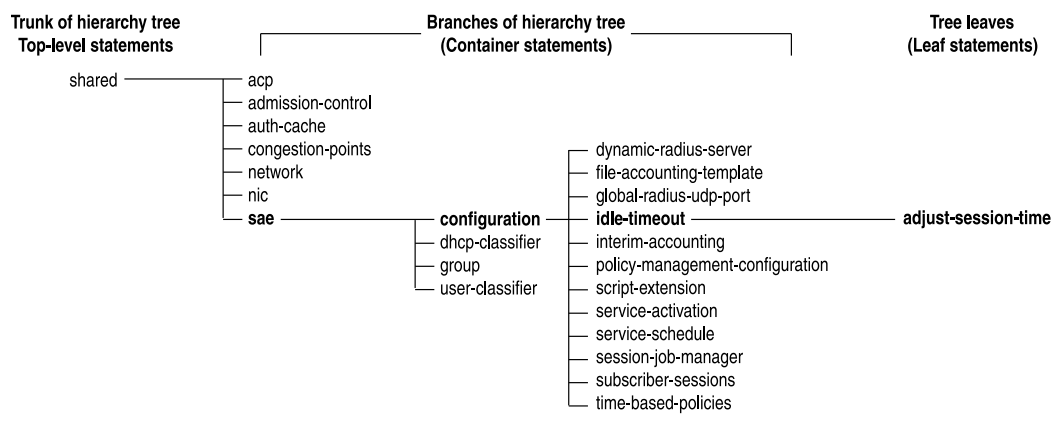
To execute a command, you enter the full command name, starting at the top level of the hierarchy. For example, to display information about SAE licenses, you enter the **show sae licenses** command.

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 on page 5 illustrates part of the hierarchy tree.

Figure 3: Configuration Statement Hierarchy Example



Related Documentation

- Overview of Commands to Control the SRC CLI Environment on page 85
- Types of SRC Commands and Statements on page 23
- SRC CLI Command Categories on page 39
- Moving Among Hierarchy Levels in the SRC CLI on page 28
- Getting Help for Commands and Statements for the SRC CLI on page 31

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). 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.

Related Documentation

- Overview of the SRC CLI on page 3
- Using SRC CLI Command Completion on page 33
- Using Keyboard Sequences at the MORE Prompt in the SRC CLI on page 71
- Getting Help for Commands and Statements for the SRC CLI on page 31

CHAPTER 2

Getting Started: A Quick Tour of the SRC CLI

- Before You Start the SRC CLI on page 7
- Starting the SRC CLI on page 8
- Displaying Commands on page 8
- Verifying System Status on page 9
- Configuring a User Account on page 11
- Creating a Configuration on page 13
- Using Shortcuts to Create a Configuration on page 15
- Making Changes to the Configuration on page 15
- Rolling Back Configuration Changes on page 16

Before You Start the SRC CLI

Make sure the SRC software has been configured for 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.

Related Documentation

- Starting the SRC CLI on page 8
- Key Features of the SRC CLI on page 5
- Overview of the SRC CLI on page 3
- For information about initial CLI configuration, see the *SRC PE 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.

To log in to a C Series Controller and start the CLI:

1. Log in to a C Series Controller through an account that has super-user privileges.



NOTE: If you enter an incorrect password, you are prompted to enter an LDAP password.

For example, to log in to a C Series Controller through an SSH session:

```
# ssh my_admin@my_cseries_platform
```

2. Start the CLI:

```
root# cli
--- SRC CLI 7.0 build CLI.B.7.0.0.006
(c) 2005-2006 Juniper Networks Inc.
user@host>
```

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

Related Documentation

- Before You Start the SRC CLI on page 7
- Displaying Commands on page 8
- Viewing Information About the SRC CLI (C-Web Interface)
- Understanding SRC Command and Statement Hierarchies on page 4
- Starting the Policies, Services, and Subscribers CLI
- For information about the SRC CLI, see Overview of the SRC CLI on page 3.

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
tracert	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
```

Related Documentation

- Getting Help for Commands and Statements for the SRC CLI on page 31
- Overview of Commands to Control the SRC CLI Environment on page 85
- Understanding SRC CLI Command Modes on page 3

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 SRC components
configuration Information about the SRC 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      myCSeriesController
Manufacturer  Juniper Networks
Product Name  C-2000
Version       1.0
Serial Number 0207082006000001
UUID          48384441-5254-0030-4859-0030485977EE
Hostid        e30a2e07
Software version SRC Release 2.0
```

```
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

Related Documentation

- Viewing C Series Controller Information on page 45
- Viewing Settings for the SRC CLI on page 90
- Viewing Information About the SRC CLI (C-Web Interface)
- Overview of Configuration for the SRC CLI

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 module, an aggregate service is a type of SAE service that comprises a number of individual services. Combining services lets the SRC module 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.
```

Related Documentation

- Using Shortcuts to Create a Configuration on page 15
- Making Changes to the Configuration on page 15
- Initially Configuring the SAE
- Displaying the Current Configuration on page 59
- Verifying a Configuration on page 68

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
```

Related Documentation

- Creating a Configuration on page 13
- Committing a Configuration on page 68
- Displaying the Current Configuration on page 59
- Verifying a Configuration on page 68

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
```

Related Documentation

- Rolling Back Configuration Changes on page 16
- Committing a Configuration on page 68
- Displaying the Current Configuration on page 59
- Verifying a Configuration on page 68
- Commands to Modify a Configuration on page 57

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.

```

**Related
Documentation**

- [Making Changes to the Configuration on page 15](#)
- [Displaying the Current Configuration on page 59](#)
- [Verifying a Configuration on page 68](#)
- [Commands to Modify a Configuration on page 57](#)

CHAPTER 3

SRC CLI Basics

- Elements of the Command-Line Interface on page 19
- SRC CLI Messages on page 20
- Displaying Command Output on page 21
- Types of SRC Commands and Statements on page 23
- Switching Between Operational Mode and Configuration Mode on page 26
- Moving Among Hierarchy Levels in the SRC CLI on page 28
- Displaying SRC CLI Command History on page 29

Elements of the Command-Line Interface

Figure 4 on page 19 shows elements of the command-line interface in operational mode.

Figure 4: Elements of the Command-Line Interface

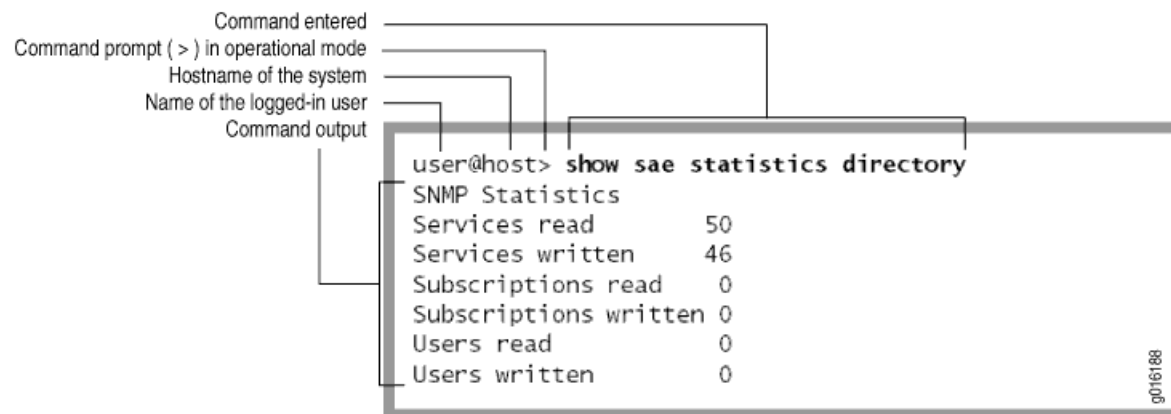


Figure 5 on page 20 shows elements of the command-line interface in configuration mode. In configuration mode, the prompt changes from a > to a #.

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 on page 20.)

Figure 6: Hierarchy-Level Banner

```

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

Related Documentation

- Overview of the SRC CLI on page 3
- Before You Start the SRC CLI on page 7
- Starting the SRC CLI on page 8
- Viewing Settings for the SRC CLI on page 90
- Overview of Configuration for the SRC CLI

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.
  
```



```
load myconfig-file<Enter>
```

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'
}
```

Related Documentation

- Moving Among Hierarchy Levels in the SRC CLI on page 28
- Using Command Completion in Configuration Mode on page 33
- Entering Configuration Mode on page 55
- Exiting from Configuration Mode on page 56
- Setting Command Completion for the SRC CLI on page 90

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 on page 22.)

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 3 on page 22 lists common keyboard sequences you can use at the **---(more)---** prompt.

Table 3: 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

Table 3: MORE Prompt Keyboard Sequences (*continued*)

Category	Action	Keyboard Sequence
	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.	-E (hyphen E)
	To exit this mode, press q .	

Related Documentation

- Displaying the End of the Output for a Command on page 78
- Displaying Output That Matches a Regular Expression on page 78
- Saving Output to a File on page 79
- Setting the Screen Length for the SRC CLI on page 88
- Using Keyboard Sequences at the MORE Prompt in the SRC CLI on page 71

Types of SRC Commands and Statements

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

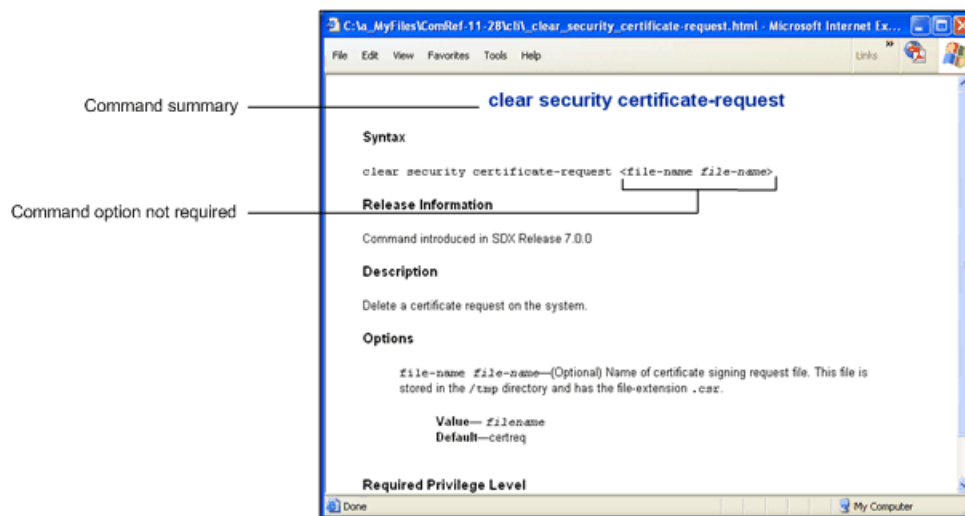
- **Operational mode commands**—Commands that you enter in operational mode are used to monitor system operation.
For more information about using top-level CLI operational mode commands, see “SRC CLI Command Categories” on page 39.
- **Environment commands**—A set of operational mode commands that you can use to control the CLI environment. For example, you can specify editing level for the CLI. For more information, see “Overview of Commands to Control the SRC CLI Environment” on page 85.
- **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 “Understanding SRC CLI Configuration Mode” on page 51.
- **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.

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 References* 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 on page 24.) Dots after an option indicate that more than one value can be supplied for the option.

Figure 8: Command Options



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 C Series Controller
> 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 **ssh** 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 “Overview of Commands to Control the SRC CLI Environment” on page 85.

Required privilege levels are listed in command and statement summaries. .

**Related
Documentation**

- Understanding SRC Command and Statement Hierarchies on page 4
- Commonly Used Operational Mode Commands on page 40
- *SRC PE CLI Command Reference*

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. Topics that describe switching modes include:

- Switching to Configuration Mode on page 26
- Returning to Operational Mode on page 26
- Running Operational Mode Commands from Configuration Mode on page 27

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;
      fallback-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

Related Documentation

- Entering Configuration Mode on page 55
- Exiting from Configuration Mode on page 56
- Displaying SRC CLI Command History on page 29
- Commonly Used Operational Mode Commands on page 40
- Types of SRC Commands and Statements on page 23

Moving Among Hierarchy Levels in the SRC CLI

You can use the CLI commands in Table 4 on page 28 to navigate the levels of the configuration statement hierarchy.

Table 4: CLI Configuration Mode Navigation Commands

Command	Description
<code>edit</code> <i>hierarchy-level</i>	Moves to an existing configuration statement hierarchy or creates a hierarchy and moves to that level.
<code>exit</code>	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.
<code>up number</code>	Moves up the hierarchy level to the specified number of levels. The up command moves to the top level if the specified number is larger than the total hierarchy levels.
<code>top</code> <i>configuration</i> <i>-command</i>	If you do not give any configuration command, the command moves directly to the top level of the configuration hierarchy. If you give a configuration command, the command is executed from the top level of the configuration hierarchy. NOTE: You cannot combine the top configuration command with exit , rollback , run , top , and up .

Related Documentation

- Creating an SRC Configuration on page 55
- Entering Configuration Mode on page 55
- Using Command Completion in Configuration Mode on page 33
- Understanding SRC Command and Statement Hierarchies on page 4

- Understanding SRC CLI Configuration Mode on page 51

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.

Related Documentation

- Displaying Commands on page 8
- Getting Help for Commands and Statements for the SRC CLI on page 31
- Types of SRC Commands and Statements on page 23
- Understanding SRC Command and Statement Hierarchies on page 4
- SRC CLI Command Categories on page 39

CHAPTER 4

Getting Online Help for the SRC CLI

- Getting Help for Commands and Statements for the SRC CLI on page 31
- Getting Help for Omitted Statements on page 32
- Using SRC CLI Command Completion on page 33
- Using Command Completion in Configuration Mode on page 33

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 SRC 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 SRC 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 SRC component.
```

Related Documentation

- Getting Help for Omitted Statements on page 32
- Overview of Commands to Control the SRC CLI Environment on page 85
- Understanding SRC Command and Statement Hierarchies on page 4
- Types of SRC Commands and Statements on page 23
- SRC CLI Messages on page 20

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}0N50d4BqvwPKY";
  }
}
```

Related Documentation

- Getting Help for Commands and Statements for the SRC CLI on page 31
- Types of SRC Commands and Statements on page 23
- Adding Configuration Statements and Identifiers on page 62
- SRC CLI Messages on page 20

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/maillogSize: 0, Last changed: Sep 27, 2006 10:33 AM
/var/log/messagesSize: 109569, Last changed: Oct 2, 2006 3:10 PM
```

Related Documentation

- Getting Help for Commands and Statements for the SRC CLI on page 31
- Using Command Completion in Configuration Mode on page 33
- Setting Command Completion for the SRC CLI on page 90
- SRC CLI Command Categories on page 39
- SRC CLI Messages on page 20

Using Command Completion in Configuration Mode

CLI command completion also applies to 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 C Series Controller
> 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

```

Related Documentation

- Using SRC CLI Command Completion on page 33
- Getting Help for Commands and Statements for the SRC CLI on page 31
- Setting Command Completion for the SRC CLI on page 90
- Understanding SRC CLI Configuration Mode on page 51

PART 2

Operational Mode and Configuration Mode

- Using the SRC CLI Operational Commands to Monitor the SRC Software on page 39
- Using Commands and Statements to Configure the SRC Software on page 51
- Filtering Command Output in the SRC CLI on page 71
- Using Keyboard Shortcuts in the SRC CLI on page 81

CHAPTER 5

Using the SRC CLI Operational Commands to Monitor the SRC Software

- SRC CLI Command Categories on page 39
- Commonly Used Operational Mode Commands on page 40
- Viewing Files and Directories on page 41
- Managing SRC Modules and Components on page 45
- Viewing Information about Users Logged Into the SRC Software on page 49

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.
 - **monitor**—Displays real-time statistics for SRC modules and components.
 - **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.
- A command—**file monitor**—for displaying online updates of a file. To stop the display, press the **Ctrl+C** keyboard sequence.

- Commands for restarting software processes—The commands in the **restart** hierarchy restart the various SRC modules and components; for example the SRC Admission Control Plug-In (SRC ACP), SNMP agent, the SRC service and policy editor, the Juniper Networks database, SRC Juniper Policy Server, the SRC network information collector, the SRC redirect server, SAE, and the C-Web interface.
- 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 modules and components.
- A command—**quit**—to exit the CLI.

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

Related Documentation

- Types of SRC Commands and Statements on page 23
- Displaying Commands on page 8
- Getting Help for Commands and Statements for the SRC CLI on page 31
- Commonly Used Operational Mode Commands on page 40
- Overview of Commands to Control the SRC CLI Environment on page 85

Commonly Used Operational Mode Commands

Table 5 on page 40 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 5: Commonly Used Operational Mode Commands

Items to Check	Description	Command
CLI	Settings for CLI environment	show cli
Configuration	Current system configuration	show configuration
	Configuration at a specific configuration level	show display level <number>
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

Table 5: Commonly Used Operational Mode Commands (*continued*)

Items to Check	Description	Command
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

Related Documentation

- Types of SRC Commands and Statements on page 23
- SRC CLI Command Categories on page 39
- Overview of Commands to Control the SRC CLI Environment on page 85
- Viewing C Series Controller Information on page 45

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. Topics include:

- Directories on the C Series Controller on page 41
- Listing Files and Directories on page 42
- Specifying Filenames and URLs on page 44

Directories on the C Series Controller

The C Series Controller has numerous directories used by the operating system. Table 6 on page 41 lists directories on a C Series Controller.

Table 6: Directories on a C Series Controller

Directory	Description
<i>/altroot</i>	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.
<i>/altvar</i>	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.
<i>/media</i>	Mount points created automatically for dynamic devices (for example, USB flash drive)
<i>/opt/UMC</i>	Files for installed components
<i>/tmp</i>	Temporary files

Table 6: Directories on a C Series Controller (*continued*)

Directory	Description
<code>/var/home</code>	Home directory for local users
<code>/var/log</code>	System log files
<code>/var/UMC</code>	Operational files and log files

Related Documentation

- Listing Files and Directories on page 42
- Specifying Filenames and URLs on page 44
- Viewing C Series Controller Information on page 45

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:
archiveArchive files from the system (local)
checksumCalculate file checksum
compareCompare files (local)
copyCopy files
deleteDelete a file (local)
listList files (local)
renameRename a file (local)
showShow 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
/.autofsckSize: 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.
. . .
```

Related Documentation

- Setting the Directory for the SRC CLI on page 89
- Specifying Filenames and URLs on page 44
- Directories on the C Series Controller on page 41
- Overview of Commands to Control the SRC CLI Environment on page 85
- Commonly Used Operational Mode Commands on page 40

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 Not in 7.0.0 Hypertext Transfer Protocol (HTTP) or 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.
```

Related Documentation

- Listing Files and Directories on page 42
- Directories on the C Series Controller on page 41
- Commonly Used Operational Mode Commands on page 40

Managing SRC Modules and Components

You use operational commands to manage SRC modules and components. Topics include:

- Viewing C Series Controller Information on page 45
- Restarting an SRC Module or Component on page 47
- Stopping the SRC Software on page 47
- Rebooting the SRC Software on page 49

Viewing C Series Controller Information

Purpose View general information about the C Series Controller, including hostname and version information for the SRC software installed on your system, and status information for the installed modules and components.

Action in operational mode, type the following command:

```
user@host>
show system information
System Identification
Hostname my C Series Controller
ManufacturerJuniper Networks
```

Product NameC-2000
Version1.0
Serial Number0207082006000001
UUID48384441-5254-0030-4859-0030485977EE
Hostid30a2e07
Software version SRC Release 2.0

System Time

Current time2006-11-30 12:09:15 EST
Uptime20 days, 13:06
Number of active users4
Load Averages (1m/5m/15m)0.19/0.22/0.18

Memory

Total15G
Free6955M

CPU Info

Number of CPU4
CPU ModelDual Core AMD Opteron(tm) Processor 265
Clock Speed1804.158 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

- Related Documentation**
- Directories on the C Series Controller on page 41
 - Verifying Status of SRC Components on page 113

Restarting an SRC Module or Component

In some instances, you may need to restart an SRC module or component. You can restart a component gracefully, immediately, or by sending a hangup signal before restarting the component. Table 7 on page 47 shows options available for the **restart** command.

Table 7: Options to Restart an SRC Module or Component

Option	Description
<i>component</i>	Name of the module or component to restart.
gracefully <i>component</i>	Restart a specified module or component by sending the equivalent of a UNIX SIGTERM signal.
immediately <i>component</i>	Immediately restart a module or component by sending the equivalent of a UNIX SIGKILL signal.
soft <i>component</i>	Reread and reactivate the configuration without completely restarting a module or 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...
```

- Related Documentation**
- Enabling SRC Components on page 113
 - Disabling an SRC Component on page 114
 - Stopping the SRC Software on page 47
 - Rebooting the SRC Software on page 49

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 nsd: [ 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
```

**Related
Documentation**

- Restarting an SRC Module or Component on page 47
- Rebooting the SRC Software on page 49

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 nsd: [ 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.
```

- Related Documentation**
- Stopping the SRC Software on page 47
 - Restarting an SRC Module or Component on page 47

Viewing Information about Users Logged Into the SRC Software

Purpose Obtain information about users currently logged into the SRC software from the SRC CLI.

Action In operational mode, type the following command:

```
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
```

Meaning In the output, the fields provide the following information:

- 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 to the 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.

Related Documentation

- [When Multiple Users Configure the Software on page 70](#)
- [Viewing Information About the SRC CLI \(C-Web Interface\)](#)

CHAPTER 6

Using Commands and Statements to Configure the SRC Software

- Understanding SRC CLI Configuration Mode on page 51
- Working in Configuration Mode with the SRC CLI on page 55
- Modifying the Configuration on page 57
- Verifying a Configuration on page 68
- Committing a Configuration on page 68
- Committing a Configuration and Exiting Configuration Mode on page 69
- When Multiple Users Configure the Software on page 70

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 modules, user access, and system properties.

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 load configuration from a file that you create. To activate the configuration, you commit it.

Configuration Mode Commands

Table 8 on page 51 summarizes each CLI configuration mode command. The commands are listed alphabetically.

Table 8: Summary of Configuration Mode Commands

Command	Description
commit	Commit the set of changes to the database and cause the changes to take operational effect.

Table 8: Summary of Configuration Mode Commands (*continued*)

Command	Description
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. With the set command, you can also set more than one option for a configuration statement.
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 9 on page 53 describes top-level CLI configuration mode statements.

Table 9: 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*.

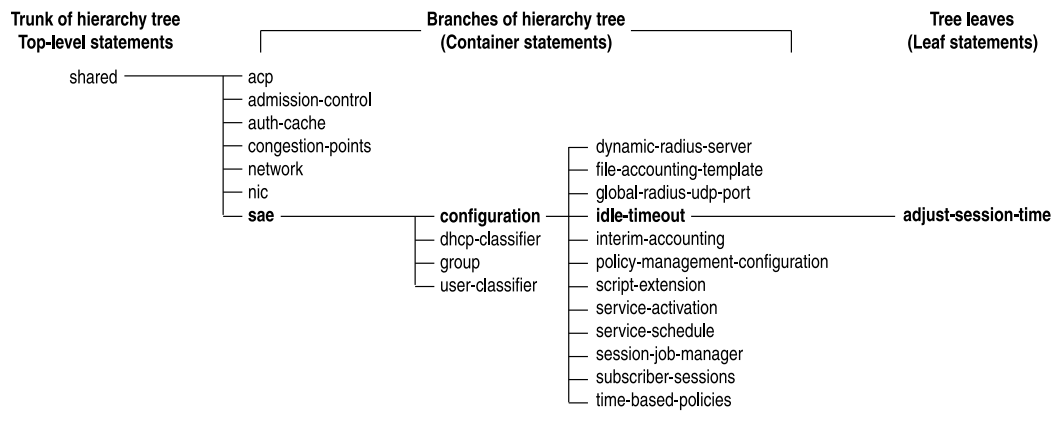
Configuration Statement Hierarchy

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 on page 54 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 on page 54 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.

**Related
Documentation**

- Replacing the Current Configuration with the Default SRC Configuration on page 104
- Merging the Active Configuration with Another Configuration on page 105
- Replacing the Configuration on page 106
- Replacing Parts of the Configuration on page 108
- Adding a Configuration Through Configuration Mode Commands on page 109
- Loading a Configuration at a Specified Hierarchy Level on page 110

Working in Configuration Mode with the SRC CLI

You configure the SRC software in configuration mode. Topics include:

- Creating an SRC Configuration on page 55
- Entering Configuration Mode on page 55
- Exiting from Configuration Mode on page 56

Creating an SRC Configuration

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.

**Related
Documentation**

- SRC Configuration Updates on page 95
- Understanding SRC CLI Configuration Mode on page 51
- Entering Configuration Mode on page 55
- Committing a Configuration and Exiting Configuration Mode on page 69
- When Multiple Users Configure the Software on page 70
- Verifying a Configuration on page 68

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
```

To enter configuration mode and lock the configuration to prevent other CLI or C-Web sessions from modifying the configuration in the database:

- Use the **configure exclusive** command.

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

Related Documentation

- [Exiting from Configuration Mode on page 56](#)
- [Understanding SRC CLI Configuration Mode on page 51](#)
- [Commands to Modify a Configuration on page 57](#)
- [Creating an SRC Configuration on page 55](#)

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>
```

**Related
Documentation**

- Entering Configuration Mode on page 55
- Committing a Configuration and Exiting Configuration Mode on page 69
- Understanding SRC CLI Configuration Mode on page 51

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. Topic include:

- Commands to Modify a Configuration on page 57
- Entering Values for Statement Options on page 58
- Displaying the Current Configuration on page 59
- Adding Configuration Statements and Identifiers on page 62
- Deleting a Statement from the Configuration on page 63
- Renaming an Identifier on page 65
- Inserting a New Identifier on page 65
- Copying a Configuration from One Configuration Location to Another on page 67
- Displaying Set Commands for the Configuration (SRC CLI) on page 67

Commands to Modify a Configuration

To modify the hierarchy, use the following 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 can also set more than one option for a configuration statement. For example:

user@host# **set transaction-variable realm operator equals value aol**

where **transaction-variable**, **operator**, and **value** are options.

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

Related Documentation

- Creating an SRC Configuration on page 55
- Making Changes to the Configuration on page 15
- Displaying the Current Configuration on page 59
- Deleting a Statement from the Configuration on page 63
- Reverting to a Previous SRC Configuration on page 111

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.

In order to distinguish parameter names from literal string values, literal string values need to be enclosed in double quotation marks. To enter them in the SRC CLI, the double quotation marks must be escaped. The recommended way to do this in the SRC CLI is to enclose the literal string (with double quotation marks) within single quotation marks—for example, "abc" is entered in the SRC CLI as "'abc'". Alternatively, you can enter the literal in double quotation marks within backslashes—for example, "\"abc\"". If just abc is used, it is considered a parameter and the SRC policy engine searches for the value of that parameter.

For convenience, where there is a fixed or well-known set of values for a parameter type, the SRC software provides a set of built-in global parameter definitions for each well-known value in the type. The SRC CLI generates completion lists for policy attributes of a given type from the set of global parameters defined for that type. You can also define your own global parameters to avoid the need to enter literal strings in quotations.

For example, a global parameter `abc="abc"` could be defined to make it convenient to enter the literal string "abc".

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:

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

Related Documentation

- Adding Configuration Statements and Identifiers on page 62
- Deleting a Statement from the Configuration on page 63
- Understanding SRC CLI Configuration Mode on page 51
- Creating an SRC Configuration on page 55

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;
      }
    }
  }
  . . .
}
}

```

Related Documentation

- Creating an SRC Configuration on page 55
- Before You Load a Configuration on page 96
- Commands to Load a Configuration on page 96
- Committing a Configuration on page 68
- Verifying a Configuration on page 68

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 C Series Controller
> 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>.

```

Related Documentation

- Understanding SRC CLI Configuration Mode on page 51
- Cutting and Pasting Configuration Information at the SRC CLI on page 112
- Renaming an Identifier on page 65
- Inserting a New Identifier on page 65
- Deleting a Statement from the Configuration on page 63

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 SRC **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

```

#### Related Documentation

- Adding Configuration Statements and Identifiers on page 62
- Understanding SRC CLI Configuration Mode on page 51
- Creating an SRC Configuration on page 55

## 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

```

#### Related Documentation

- Inserting a New Identifier on page 65
- Adding Configuration Statements and Identifiers on page 62
- Understanding SRC CLI Configuration Mode on page 51
- Creating an SRC Configuration on page 55

## 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 != "";
}
}
```

#### Related Documentation

- Renaming an Identifier on page 65
- Adding Configuration Statements and Identifiers on page 62
- Understanding SRC CLI Configuration Mode on page 51
- Creating an SRC Configuration on page 55

## Copying a Configuration from One Configuration Location to Another

You can copy a collection of configuration statements from one place in the configuration to another. This process simplifies configuration so that you do not need to configure the same information in more than one place.

To copy a collection of configuration statements from one location to another:

- `user@host# copy <source edit path> <source identifier name>to <target edit path> <target identifier name>`

For example, to copy the configuration for user Chris to another use Pat:

```
[edit system login]
user@host# copy user Chris to user Pat
```

#### Related Documentation

- Displaying the Current Configuration on page 59
- Cutting and Pasting Configuration Information at the SRC CLI on page 112
- Reverting to a Previous SRC Configuration on page 111
- Committing a Configuration on page 68
- How the SRC Configuration Is Stored on page 93

## Displaying Set Commands for the Configuration (SRC CLI)

You can display the candidate configuration using the **display set** command along with the **show** command. The configuration displayed can be used as the basis for another configuration at the top level of the configuration hierarchy or at the current hierarchy level.

To display the configuration in the form of the **set** command:

```
[edit system login]
user@host# show user admin | display set <relative>
```

The following example displays the **show user admin | display set** command output for the set configuration:

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

```
set system login user admin class super-user
set system login user admin full-name admin
set system login user admin uid 500
set system login user admin gid 100
set system login user admin authentication encrypted-password
"{crypt}.42km6cWKhX72"
set system login user admin level normal
set system login user admin complete-on-space on
```

- Related Documentation**
- How the SRC Configuration Is Stored on page 93
  - Understanding SRC CLI Configuration Mode on page 51
  - Creating an SRC Configuration on page 55

---

## Verifying a Configuration

**Purpose** Verify that the syntax of a configuration is correct.

**Action** 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'
```

- Related Documentation**
- Making Changes to the Configuration on page 15
  - Committing a Configuration on page 68
  - Displaying the Current Configuration on page 59
  - Verifying the Configuration for SRC Directory Access
  - Creating an SRC Configuration on page 55

---

## 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.

#### Related Documentation

- [Displaying the Current Configuration on page 59](#)
- [Copying a Configuration from One Configuration Location to Another on page 67](#)
- [Exiting from Configuration Mode on page 56](#)
- [Verifying a Configuration on page 68](#)
- [Creating an SRC Configuration on page 55](#)

---

## 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>
```

- Related Documentation**
- Committing a Configuration on page 68
  - Exiting from Configuration Mode on page 56
  - Entering Configuration Mode on page 55
  - Verifying a Configuration on page 68
  - Creating an SRC Configuration on page 55

---

## 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>
```

- Related Documentation**
- Commands to Modify a Configuration on page 57
  - Entering Configuration Mode on page 55
  - Viewing Information about Users Logged Into the SRC Software on page 49
  - Creating an SRC Configuration on page 55
  - How the SRC Configuration Is Stored on page 93

## CHAPTER 7

# Filtering Command Output in the SRC CLI

- Using Keyboard Sequences at the MORE Prompt in the SRC CLI on page 71
- Using the Pipe ( | ) Symbol When Entering Commands on page 73
- Using Regular Expressions with the Pipe Symbol on page 73
- Using Pipe Filter Functions on page 74

### 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 10 on page 71 lists the keyboard sequences for the commands most frequently used at the **MORE** prompt.

**Table 10: MORE Prompt Keyboard Sequences**

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

Table 10: MORE Prompt Keyboard Sequences (*continued*)

| Category                                                                  | Action                                                                                                                                                                                       | Keyboard Sequence                      |
|---------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
|                                                                           | Jump to the last line in the 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                                      |
| 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.                                                                                                                                                                 | /string                                |
|                                                                           | You can also search for a string by specifying the <b> match</b> filter when entering a command using the pipe symbol. See “Displaying Output That Matches a Regular Expression” on page 78. |                                        |
|                                                                           | 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                                      |

**Related Documentation**

- Using the Pipe ( | ) Symbol When Entering Commands on page 73
- Saving Output to a File on page 79

## 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 C Series Controller
 > 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
```

### Related Documentation

- Using Regular Expressions with the Pipe Symbol on page 73
- Displaying the End of the Output for a Command on page 78
- Preventing Output from Being Paginated on page 79
- Displaying Output in XML Tag Format on page 76
- Saving Output to a File on page 79

## 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 11 on page 74.) If the regular expression contains spaces, operators, or wildcard characters, enclose the expression in quotation marks.

**Table 11: 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 Documentation

- Using the Pipe ( | ) Symbol When Entering Commands on page 73
- Disregarding Output That Does Not Match a Regular Expression on page 76
- Displaying Output from the First Match of a Regular Expression on page 77
- Displaying Output That Matches a Regular Expression on page 78
- Saving Output to a File on page 79

## Using Pipe Filter Functions

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

- Counting the Number of Lines of Output on page 75
- Displaying Summarized Output for a Hierarchy Level (SRC CLI) on page 75
- Displaying Output in XML Tag Format on page 76

- Disregarding Output That Does Not Match a Regular Expression on page 76
- Displaying Output from the First Match of a Regular Expression on page 77
- Displaying the End of the Output for a Command on page 78
- Displaying Output That Matches a Regular Expression on page 78
- Preventing Output from Being Paginated on page 79
- Saving Output to a File on page 79

## 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
```

### Related Documentation

- Displaying Output in XML Tag Format on page 76
- Displaying the End of the Output for a Command on page 78
- Preventing Output from Being Paginated on page 79
- Saving Output to a File on page 79
- Using the Pipe ( | ) Symbol When Entering Commands on page 73

## Displaying Summarized Output for a Hierarchy Level (SRC CLI)

To display the summarized output of the **display level** command along with the **show** command:

```
[edit shared]
user@host# show | display level <level>
```



**NOTE:** The level filter specifies the configuration level for which the summarized output is displayed.

The following example displays the **show | display level 1** command output for the hierarchy level 1:

```
[edit shared]
user@host# show | display level 1

acp;
auth-cache;
network;
nic;
sae;
```

### Related Documentation

- Saving Output to a File on page 79

- [Displaying Output in XML Tag Format on page 76](#)
- [Understanding SRC Command and Statement Hierarchies on page 4](#)
- [Using the Pipe \( | \) Symbol When Entering Commands on page 73](#)

## 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>
```

### Related Documentation

- [Displaying Command Output on page 21](#)
- [Counting the Number of Lines of Output on page 75](#)
- [Displaying the End of the Output for a Command on page 78](#)
- [Saving Output to a File on page 79](#)
- [Using the Pipe \( | \) Symbol When Entering Commands on page 73](#)

## 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 11 on page 74.

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;
 }
 }
}
```



```
 }
 }
}
```

#### Related Documentation

- Displaying Output from the First Match of a Regular Expression on page 77
- Displaying Output That Matches a Regular Expression on page 78
- Saving Output to a File on page 79
- Using Regular Expressions with the Pipe Symbol on page 73

## 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 11 on page 74.

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%
```

```
. . .
```

#### Related Documentation

- Disregarding Output That Does Not Match a Regular Expression on page 76
- Displaying Output That Matches a Regular Expression on page 78
- Saving Output to a File on page 79
- Using Regular Expressions with the Pipe Symbol on page 73

## 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
```

### Related Documentation

- Counting the Number of Lines of Output on page 75
- Displaying Output in XML Tag Format on page 76
- Preventing Output from Being Paginated on page 79
- Saving Output to a File on page 79
- Using the Pipe ( | ) Symbol When Entering Commands on page 73

## 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 11 on page 74.

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 {
```

### Related Documentation

- Disregarding Output That Does Not Match a Regular Expression on page 76
- Displaying Output from the First Match of a Regular Expression on page 77
- Saving Output to a File on page 79
- Using Regular Expressions with the Pipe Symbol on page 73

## 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
```

### Related Documentation

- Displaying Output in XML Tag Format on page 76
- Counting the Number of Lines of Output on page 75
- Displaying the End of the Output for a Command on page 78
- Saving Output to a File on page 79

## 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>
```

### Related Documentation

- Counting the Number of Lines of Output on page 75
- Displaying Output in XML Tag Format on page 76
- Preventing Output from Being Paginated on page 79

- [Using the Pipe \( | \) Symbol When Entering Commands on page 73](#)

## 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 12 on page 81 lists some of the CLI keyboard sequences. They are the same as those used in Emacs.

**Table 12: CLI Keyboard Sequences**

Category	Action	Keyboard Sequence
Move the Cursor	Move the cursor back one character.	<b>Ctrl+b</b>
	Move the cursor back one word.	<b>Esc+b</b> or <b>Alt+b</b>
	Move the cursor forward one character.	<b>Ctrl+f</b>
	Move the cursor forward one word.	<b>Esc+f</b> or <b>Alt+f</b>
	Move the cursor to the beginning of the command line.	<b>Ctrl+a</b>
	Move the cursor to the end of the command line.	<b>Ctrl+e</b>

Table 12: CLI Keyboard Sequences (*continued*)

Category	Action	Keyboard Sequence
Delete Characters	Delete the character before the cursor.	<b>Delete or Backspace</b>
	Delete the character at the cursor.	<b>Ctrl+d</b>
	Delete all characters from the cursor to the end of the command line.	<b>Ctrl+k</b>
	Delete all characters on the command line.	<b>Ctrl+u or Ctrl+x</b>
	Delete the word before the cursor.	<b>Ctrl+w, Esc+Backspace, Alt+Delete, or Alt+Backspace</b>
	Delete the word after the cursor.	<b>Esc+d or Alt+d</b>
Insert Recently Deleted Text	Insert the most recently deleted text at the cursor.	<b>Ctrl+y</b>
Redraw the Screen	Redraw the current line.	<b>Ctrl+l</b>
Display Previous Command Lines	Scroll backward through the list of recently executed commands.	<b>Ctrl+p or up arrow</b>
	Scroll forward through the list of recently executed commands.	<b>Ctrl+n or down arrow</b>
	Search the CLI history in reverse order for lines matching the search string.	<b>Ctrl+r</b>
		<b>Ctrl+s</b>
	Terminate a search, and display the found command.	<b>Esc or Ctrl+j</b>
	Abort a search.	<b>Ctrl+g</b>
Repeat Keyboard Sequences	Specify the number of times to execute a keyboard sequence; <i>number</i> can be from 1 through 9.	<b>Esc+ <i>number sequence</i> or Alt+ <i>number sequence</i></b>

## PART 3

# SRC CLI Environment and SRC Module and Component Management

- Controlling the SRC CLI Environment on page 85
- Managing SRC Configurations on page 93
- Managing SRC Modules and Components with the CLI on page 113





## CHAPTER 9

# Controlling the SRC CLI Environment

- Overview of Commands to Control the SRC CLI Environment on page 85
- Setting the Editing Level for the SRC CLI on page 86
- Setting the Terminal Type for the SRC CLI on page 87
- Setting the Language for the Terminal Environment for the SRC CLI on page 87
- Setting the Screen Length for the SRC CLI on page 88
- Setting the Screen Width for the SRC CLI on page 88
- Changing the Password for the SRC CLI on page 88
- Setting the SRC CLI Prompt on page 89
- Setting the Directory for the SRC CLI on page 89
- Setting Command Completion for the SRC CLI on page 90
- Viewing Settings for the SRC CLI on page 90

### 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.

**Related Documentation**

- Setting the Editing Level for the SRC CLI on page 86
- Setting Command Completion for the SRC CLI on page 90
- Changing the Password for the SRC CLI on page 88
- Viewing C Series Controller Information on page 45
- Commonly Used Operational Mode Commands on page 40

---

## 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 13 on page 86 describes the editing levels.

**Table 13: 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.

**Related Documentation**

- Types of SRC Commands and Statements on page 23
- Overview of Commands to Control the SRC CLI Environment on page 85
- Setting the Terminal Type for the SRC CLI on page 87
- Setting the SRC CLI Prompt on page 89
- Viewing Information About the SRC CLI (C-Web Interface)

---

## 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**.

**Related Documentation**

- Setting the Language for the Terminal Environment for the SRC CLI on page 87
- Setting the Screen Length for the SRC CLI on page 88
- Setting the Screen Width for the SRC CLI on page 88
- Viewing Settings for the SRC CLI on page 90

---

## 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).

**Related Documentation**

- Setting the Terminal Type for the SRC CLI on page 87
- Setting the Screen Length for the SRC CLI on page 88
- Setting the Screen Width for the SRC CLI on page 88
- Setting the Editing Level for the SRC CLI on page 86

## 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`

### Related Documentation

- Setting the Screen Width for the SRC CLI on page 88
- Setting the Language for the Terminal Environment for the SRC CLI on page 87
- Setting the Terminal Type for the SRC CLI on page 87
- Viewing Settings for the SRC CLI on page 90
- Setting the Editing Level for the SRC CLI on page 86

## 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`

### Related Documentation

- Setting the Screen Length for the SRC CLI on page 88
- Setting the Terminal Type for the SRC CLI on page 87
- Setting the Language for the Terminal Environment for the SRC CLI on page 87
- Viewing Settings for the SRC CLI on page 90
- Setting the Editing Level for the SRC CLI on page 86

## 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.

**Related Documentation**

- Setting the Terminal Type for the SRC CLI on page 87
- Setting the SRC CLI Prompt on page 89
- Overview of Commands to Control the SRC CLI Environment on page 85
- Setting the Editing Level for the SRC CLI on page 86

## 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.

**Related Documentation**

- Starting the SRC CLI on page 8
- Switching Between Operational Mode and Configuration Mode on page 26
- Understanding SRC CLI Command Modes on page 3

## Setting the Directory for the SRC CLI

By default, on a C Series Controller the working directory is the home directory of the user. 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.

**Related Documentation**

- Changing Access to the Directory that Stores SRC Configuration Data
- Configuration Statements for SRC CLI Directory Access

- Verifying the Configuration for SRC Directory Access

## 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
```

### Related Documentation

- Using SRC CLI Command Completion on page 33
- Using Command Completion in Configuration Mode on page 33

## Viewing Settings for the SRC CLI

---

**Purpose** View the current CLI settings.

**Action** In operational mode, enter the following command:

```
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' loginName 'user' 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

```

The **show cli authorization class *class*** command can display two different outputs.

If your loginName and the name of your user profile are the same, the output is: **Current user: 'loginName'**. For example:

```

user@host>
show cli authorization
Current user: 'root' class 'super-user'

```

If your loginName and the name of your user profile differ, both names are included in the output:

**Current user: '*profile*' loginName *loginName* class *class***

The loginName displayed is the name you entered when you logged in. For example:

```

user@host>
show cli authorization
Current user: 'root' loginName 'user' class 'super-user'

```

#### Related Documentation

- Setting the Editing Level for the SRC CLI on page 86
- Setting the Terminal Type for the SRC CLI on page 87
- Setting the Screen Length for the SRC CLI on page 88
- Setting the Screen Width for the SRC CLI on page 88
- Viewing Information About the SRC CLI (C-Web Interface)





## CHAPTER 10

# Managing SRC Configurations

- How the SRC Configuration Is Stored on page 93
- Commands to Save a Configuration to a File on page 94
- Updating the SRC Configuration on page 95
- About SRC Configuration Files in XML Format on page 99
- About SRC Configuration Files in Text Format on page 101
- Preparing a File to Be Loaded into the Current SRC Configuration on page 103
- Loading an SRC Configuration on page 104
- Comparing SRC Configurations on page 110
- Reverting to a Previous SRC Configuration on page 111
- Cutting and Pasting Configuration Information at the SRC CLI on page 112

### How the SRC Configuration Is Stored

---

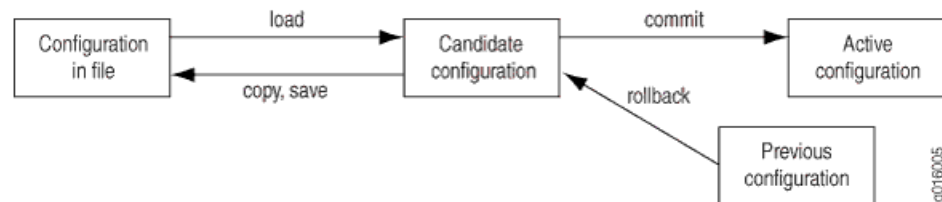
When you edit a configuration, you work in a copy of the current configuration to create a candidate configuration. Changes you make to the candidate configuration are visible in the CLI immediately. If multiple users edit the configuration at the same time, all users can see all changes.

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.

You can save the local (slot) configuration and the shared configuration. The slot (local) configuration is stored in either a text file or XML file, and the remainder of the configuration (shared) is stored in the Juniper Networks database or another directory that you have configured to store SRC configuration data.

Figure 10 on page 94 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



#### Related Documentation

- Creating an SRC Configuration on page 55
- SRC Configuration Updates on page 95
- About SRC Configuration Files in XML Format on page 99
- About SRC Configuration Files in Text Format on page 101
- Cutting and Pasting Configuration Information at the SRC CLI on page 112

## Commands to Save a Configuration to a File

You can save both the local and shared configuration of the C Series Controller. The local (slot) configuration is configuration data that is stored in a C Series Controller's local file system. Shared configuration is configuration data that is stored in the Juniper Networks database. The Juniper Networks database can be in standalone mode or community mode. SSH host keys and the SRC component states (enable/disable) are automatically saved together with other configuration data.

Saving and loading the configuration allows you to fully restore a C Series Controller to the configuration and operational state based on which configuration you save and load. For example, you can restore a system to the factory-default, reinstall a system, or deploy a new system configuration from the saved configuration.

Saving and loading the local configuration allows you to restore a C Series Controller host that uses the Juniper Networks database community for the source of the shared configuration. Saving and loading a shared configuration can be used to restore C Series Controller hosts that have the Juniper Networks database set to standalone mode. In certain cases, it can also be used to restore systems running in a Juniper Networks database community.

You can use the following commands in configuration mode to save the configuration to a text or XML file:

- **save**—Saves the entire C Series Controller configuration.
- **save local**—Saves the configuration data that is stored in a C Series Controller's host's local file system—for example, those stored in properties files, XML files, and SSH host keys.
- **save shared**—Saves the configuration data that is stored in Juniper Networks database servers.

You can save the configuration data in either a text or XML file.

## SSH Host Keys

When you execute the **save local** or **save** command at the top or system edit level, SSH host keys are saved and visible in the configuration file under “system ssh-host-keys.” You cannot set or view the configuration of SSH host keys using the SRC CLI. SSH host keys are obtained from system files located in the */etc/ssh* directory. The SSH host key data saved into configuration files is refreshed with the most recent data in the files when you execute the **save** command.

## Component State

When you execute the **save local** or **save** command, the state of the SRC components is polled and saved in the configuration file under “component-state.” However, like the SSH host keys, you cannot view or set the values for the component state using the CLI.

When you load and commit the configuration from a saved file, the SRC software sets the state of the components in the C Series Controller to the state saved in the configuration file. Components that were in either the “running” or “stopped” state when polled are automatically enabled, and components that were “disabled” when polled are automatically disabled.

- Related Documentation**
- How the SRC Configuration Is Stored on page 93
  - Updating the SRC Configuration on page 95

---

## Updating the SRC Configuration

You can update an SRC configuration with information stored in a file. Topics include:

- SRC Configuration Updates on page 95
- Before You Load a Configuration on page 96
- Commands to Load a Configuration on page 96
- Attributes in SRC Configuration Files on page 98

## SRC Configuration Updates

You can update an SRC configuration to include configuration changes from a file or to revert to the configuration supplied with the 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.

- Related Documentation**
- How the SRC Configuration Is Stored on page 93
  - Attributes in SRC Configuration Files on page 98
  - Replacing the Current Configuration with the Default SRC Configuration on page 104
  - Reverting to a Previous SRC Configuration on page 111

## 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*
- XML format—*SRC 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
```

### Related Documentation

- Cutting and Pasting Configuration Information at the SRC CLI on page 112
- About SRC Configuration Files in XML Format on page 99
- About SRC Configuration Files in Text Format on page 101
- Commands to Load a Configuration on page 96
- Preparing a File to Be Loaded into the Current SRC Configuration on page 103

## Commands to Load a Configuration

Use the **load** commands to load your configuration from a saved configuration file.

The **load merge** and **load replace** commands include options for loading only system bootstrap or none-bootstrap configuration data from the specified configuration file. Bootstrap configuration data is part of the local configuration. It enables an SRC host to be a networked host, and includes parameters like system host-name, system domain-name, interfaces, and routing options.

- Use the **bootstrap-only** option when loading the configuration on C Series Controllers that have no bootstrap configuration—for example, a new system or a system that has had the factory default configuration restored.
- Use the **except-bootstrap** option when you are loading a configuration from a file and you want to retain the existing bootstrap configuration.

You can automatically enable SRC components when the CLI starts. For example, this allows you to automatically enable the Juniper Networks database community on a system that has only the factory default configuration loaded. To configure the system to automatically enable components, add “Component.auto-enable = component-name” to the CLI backend bootstrap properties file “cli.properties.”

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

- **load factory-default**—Replaces the existing configuration with the configuration supplied with the SRC software.
- **load merge (*filename* | terminal) <relative>**—Combines the configuration that is currently shown in the CLI and the configuration in the specified file or in the text you type at the terminal. Press Ctrl+D to end terminal input.
  - **bootstrap-only**—When used with this option, only system bootstrap configuration data is merged.
  - **except-bootstrap**—When used with this option, all configuration data except the bootstrap configuration is merged.
- **load override (*filename* | terminal)**—Discards the entire configuration that is currently shown in the CLI, and loads the entire configuration in the specified file or in the text you type at the terminal. Press Ctrl+D to end terminal input.
- **load replace (*filename* | terminal) <relative>**—Looks for replace attributes in the specified file, deletes the existing statements of the same name, and replaces them with the configuration in the specified file or in the text you type at the terminal. Press Ctrl+D to end terminal input.
  - **bootstrap-only**—When used with this option, only system bootstrap configuration data is replaced.
  - **except-bootstrap**—When used with this option, all configuration data except the bootstrap configuration is replaced.
- **load set (*filename* | terminal) <relative>**—Executes configuration mode commands such as **set**, **edit**, **exit**, and **top** from a text file or from the text you type at the terminal. Press Ctrl+D to end terminal input.

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.

---

### SSH Host Keys

You can load SSH host keys into the configuration from the saved configuration file. When you execute **commit**, the loaded SSH host keys are used to update system key files (those in the `/etc/ssh` directory). The original files are backed up with the same filename with an additional “~”.

---

### Component State

You can load SRC component states into the configuration from the saved configuration file. When you execute **commit**, the loaded component state values are used to automatically enable and disable the SRC components.

#### Related Documentation

- Creating an SRC Configuration on page 55
- Commands to Save a Configuration to a File on page 94
- Before You Load a Configuration on page 96
- Preparing a File to Be Loaded into the Current SRC Configuration on page 103
- Loading a Configuration at a Specified Hierarchy Level on page 110
- Cutting and Pasting Configuration Information at the SRC CLI on page 112

## 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.

#### Related Documentation

- About SRC Configuration Files in XML Format on page 99
- About SRC Configuration Files in Text Format on page 101

- Preparing a File to Be Loaded into the Current SRC Configuration on page 103

## 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 l.*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>
```

#### Related Documentation

- [Attributes in SRC Configuration Files on page 98](#)
- [About SRC Configuration Files in Text Format on page 101](#)
- [Preparing a File to Be Loaded into the Current SRC Configuration on page 103](#)

### 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 through 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 ;
 }
 }
 }
}
```

```
}
}
```

#### Related Documentation

- Attributes in SRC Configuration Files on page 98
- About SRC Configuration Files in XML Format on page 99
- Preparing a File to Be Loaded into the Current SRC Configuration on page 103

## 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 “Specifying Filenames and URLs” on page 44.

- 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>
```

#### Related Documentation

- Before You Load a Configuration on page 96
- About SRC Configuration Files in XML Format on page 99
- About SRC Configuration Files in Text Format on page 101
- Commands to Load a Configuration on page 96
- For information about enabling SSH and Telnet on the C Series Controller, see Enabling Remote Users to Access the C-Web Interface.

---

## 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 104
- Merging the Active Configuration with Another Configuration on page 105
- Replacing the Configuration on page 106
- Replacing Parts of the Configuration on page 108
- Adding a Configuration Through Configuration Mode Commands on page 109
- Loading a Configuration at a Specified Hierarchy Level on page 110

### 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.

**Related Documentation**

- Merging the Active Configuration with Another Configuration on page 105
- Replacing the Configuration on page 106
- Replacing Parts of the Configuration on page 108
- Adding a Configuration Through Configuration Mode Commands on page 109
- Reverting to a Previous SRC Configuration on page 111

## 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 110.

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>
```

#### Related Documentation

- [Replacing the Current Configuration with the Default SRC Configuration on page 104](#)
- [Adding a Configuration Through Configuration Mode Commands on page 109](#)
- [Reverting to a Previous SRC Configuration on page 111](#)

## 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>
```

#### Related Documentation

- Replacing the Current Configuration with the Default SRC Configuration on page 104
- Merging the Active Configuration with Another Configuration on page 105
- Replacing Parts of the Configuration on page 108
- Adding a Configuration Through Configuration Mode Commands on page 109
- Reverting to a Previous SRC Configuration on page 111

## 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 110.

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 “About SRC Configuration Files in XML Format” on page 99.

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 “About SRC Configuration Files in XML Format” on page 99.

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>
```

#### Related Documentation

- [Replacing the Current Configuration with the Default SRC Configuration on page 104](#)
- [Merging the Active Configuration with Another Configuration on page 105](#)
- [Replacing the Configuration on page 106](#)
- [Adding a Configuration Through Configuration Mode Commands on page 109](#)
- [Reverting to a Previous SRC Configuration on page 111](#)

## 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
```

**Related Documentation**

- Replacing the Current Configuration with the Default SRC Configuration on page 104
- Merging the Active Configuration with Another Configuration on page 105
- Replacing the Configuration on page 106
- Replacing Parts of the Configuration on page 108
- Reverting to a Previous SRC Configuration on page 111

## 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
```

**Related Documentation**

- Before You Load a Configuration on page 96
- Adding a Configuration Through Configuration Mode Commands on page 109
- Commands to Load a Configuration on page 96

## Comparing SRC Configurations

---

This sample procedure describes how to use the **compare** command to display the uncommitted changes against the active configuration.

To compare uncommitted configuration changes:

- Use the **show | compare** command to compare any uncommitted changes made at the CLI with the active configuration.

```
[edit]
user@host# show | compare
```

- Use the **show | compare filename** command to compare the candidate configuration against a text file containing the configuration. The candidate configuration is a combination of the active configuration and the uncommitted configuration changes.

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



**NOTE:** The **compare** command works only with the text format display.

The following example displays the **show | compare** command output, if the edit level of the *admin* user is changed from *normal* to *expert* and a new user *guest* is configured:

```
[edit system login]
user@host# show | compare

+ user guest {
+ class super-user;
+ full-name guest;
+ uid 501;
+ gid 100;
+ authentication {
+ encrypted-password "{crypt}jWAE2tHU5EUK";
+ }
+ level normal;
+ complete-on-space on;
+ }

[edit system login user admin]
user@host# show | compare

- level normal
+ level expert
```



**NOTE:** The plus sign (+) before the statement name indicates new configuration and the minus sign (–) before the statement name indicates deleted configuration.

#### Related Documentation

- Creating an SRC Configuration on page 55
- Types of SRC Commands and Statements on page 23
- How the SRC Configuration Is Stored on page 93
- Verifying a Configuration on page 68

## 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
```

**Related Documentation**

- Replacing the Current Configuration with the Default SRC Configuration on page 104
- Replacing the Configuration on page 106
- Replacing Parts of the Configuration on page 108
- Adding a Configuration Through Configuration Mode Commands on page 109
- Preparing a File to Be Loaded into the Current SRC Configuration on page 103

---

## 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.

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

To load a configuration from the text that you type at the terminal:

- In configuration mode, specify the **load (merge | override | replace | set) terminal** command. For example:

```
user@host# load merge terminal
```

Copy and paste configuration text from another source, such as the output of the **show** command for a configuration. Press Ctrl+D to end terminal input and load the configuration.

**Related Documentation**

- Merging the Active Configuration with Another Configuration on page 105
- Replacing the Configuration on page 106
- Replacing Parts of the Configuration on page 108
- Adding a Configuration Through Configuration Mode Commands on page 109
- Loading a Configuration at a Specified Hierarchy Level on page 110
- How the SRC Configuration Is Stored on page 93
- Commands to Load a Configuration on page 96

## CHAPTER 11

# Managing SRC Modules and Components with the CLI

- Verifying Status of SRC Components on page 113
- Enabling SRC Components on page 113
- Disabling an SRC Component on page 114
- Restarting an SRC Component on page 115

### Verifying Status of SRC Components

---

**Purpose** View information about the status for modules and components.

**Action** 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

- Related Documentation**
- Viewing Information About Components Installed (SRC CLI)
  - Verifying the Local Configuration for a Component
  - Local Properties for SRC Components
  - Configuring SRC Components

### Enabling SRC Components

---

On a C Series Controller, you can enable all SRC components from the CLI.

You can enable the following components from the CLI:

- Admission Control Plug-In (ACP)
- Service activation engine (SAE)
- Subscriber information collector (SIC)
- 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 *component*** command. For example:

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

#### Related Documentation

- Disabling an SRC Component on page 114
- Restarting an SRC Module or Component on page 47
- Configuring SRC Components

---

## Disabling an SRC Component

On a C Series Controller, you can disable a module or component that is running from 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
```

**Related  
Documentation**

- Enabling SRC Components on page 113
- Restarting an SRC Module or Component on page 47
- Configuring SRC Components

---

## 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 a module or 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
```

**Related  
Documentation**

- Enabling SRC Components on page 113
- Disabling an SRC Component on page 114
- Configuring SRC Components





## PART 4

# Index

- Index on page 119



# Index

## Symbols

* in completion list.....	25
+ in completion list.....	25
+* in completion list.....	25
> in completion list.....	25
(pipe) filter for command output.....	73

## A

active configuration.....	4
---------------------------	---

## C

C Series Controllers	
directories.....	41, 42
files.....	42
C Series controllers	
status.....	45
C Series platforms	
version.....	45
candidate configuration.....	4
CLI	
command hierarchy.....	4
command history.....	29
command mode.....	3
configuration mode	
command completion.....	33
commands.....	51
comparing.....	110
entering.....	55
exiting.....	56
hierarchy.....	28, 53, 54
navigation commands.....	28
overview.....	3, 51
statement description.....	53
symbols in statement lists.....	25, 62
verifying configurations.....	68
editing level.....	86
environment settings.....	90
help.....	8, 31
interface elements.....	19
keyboard sequences.....	81

messages.....	20
modes	
changing.....	26, 27
overview.....	3
operating systems.....	3
overview.....	3, 6
password.....	88
prerequisites.....	7
privilege levels.....	25
prompt strings.....	89
tutorial.....	7
working directory.....	89
command output	
MORE prompt.....	21
number of lines.....	75
pagination.....	79
saving to a file.....	79
See also regular expressions.....	77
XML format, displaying.....	76
commands	
categories.....	39
completion.....	33, 90
configuration mode .....	51
filenames in.....	44
operational mode.....	40
options.....	24
types.....	23
URLs in.....	44
commit and-quit command.....	26, 69
commit check command.....	68
commit command.....	51, 68
configuration	
backup.....	96
changes by multiple users.....	70
committing.....	26, 68, 69
default.....	104
deleting statements.....	63
displaying current configuration.....	59
example.....	54
loading.....	96, 109, 110
local.....	93

merge.....	105	<b>F</b>	
modifying.....	57	file commands.....	42
replacement.....	106, 108	files	
rollback.....	111	saving command output to.....	79
saving.....	94	specifying names in commands.....	44
slot.....	93	find filter.....	77
statements. <i>See</i> configuration statements		<b>H</b>	
storage.....	93	help command.....	52
verifying.....	68	history command.....	52
configuration files		history, CLI commands.....	29
attributes.....	98	<b>I</b>	
overview.....	97	identifiers	
preparation to load.....	103	inserting in sequential lists.....	65
text format.....	101	renaming.....	65
XML format.....	99	insert command.....	52, 65
configuration hierarchy.....	5	<b>K</b>	
configuration mode <i>See</i> CLI		keyboard sequences at MORE prompt.....	71
configuration statements		<b>L</b>	
deleting.....	63	load command.....	52
overview.....	24, 62	load factory-default command.....	104
symbols in statements.....	25	load merge command.....	105
configure command.....	55	load override command.....	106
conventions		load replace command.....	108
notice icons.....	xvii	load set command.....	102, 109, 110
text.....	xvii	<b>M</b>	
count filter.....	75	manuals	
cursor, moving.....	81	comments on.....	xix
customer support.....	xix	messages	
contacting JTAC.....	xix	CLI.....	20
<b>D</b>		MORE filter.....	79
delete command.....	52, 63	MORE prompt.....	71
directories on C Series Controller.....	41	<b>N</b>	
documentation		notice icons.....	xvii
comments on.....	xix	<b>O</b>	
<b>E</b>		online help <i>See</i> CLI, help	
edit command.....	52, 57	operational mode.....	3
enable command.....	114	<b>P</b>	
environment settings, CLI.....	90	privilege levels for CLI.....	25
command completion.....	90	prompt strings for CLI.....	89
editing level.....	86	<b>Q</b>	
prompt string.....	89	quit command.....	52
screen dimensions.....	88		
terminal language.....	87		
terminal type.....	87		
working directory.....	89		
except filter.....	76		
exit command.....	26, 52		

**R**

redrawing screen.....	82
regular expressions	
displaying first match.....	77
displaying matching output.....	78
disregarding nonmatching output.....	76
overview.....	73
rename command.....	52, 65
request system halt command.....	48
request system reboot.....	49
restart command.....	47, 115
rollback command.....	16, 52
run command.....	52

**S**

save command.....	52
screen	
dimensions.....	88
redrawing.....	82
set cli complete-on-space command.....	90
set cli directory command.....	42, 89
set cli language command.....	87
set cli level command.....	86
set cli password command.....	88
set cli prompt command.....	89
set cli screen-length command.....	88
set cli screen-width command.....	88
set cli terminal command.....	87
set command.....	52, 58
show cli authorization command.....	90
show cli command.....	90
show cli history command.....	29
show cli level command.....	87
show command.....	52
show component command.....	46, 113
show configuration command.....	59
show system information command.....	45
SRC CLI	
starting	
C Series platform.....	8
SRC components	
disabling.....	114
enabling.....	113
managing.....	45
restarting.....	47, 115
status.....	45
SRC modules	
managing.....	45
restarting.....	47

SRC software	
rebooting.....	49
stopping.....	47
statements See configuration statements, names	
of individual statements	
support, technical See technical support	
symbols in statements.....	25

**T**

technical support	
contacting JTAC.....	xix
terminal type.....	87
text conventions defined.....	xvii
top command.....	52

**U**

up command.....	52
URLs in commands.....	44
user accounts	
configuration example.....	11

**W**

working directory.....	89
------------------------	----

**X**

XML format, command output in.....	76
------------------------------------	----

