



SRC PE Software

CLI User Guide

Release

4.13.x



Modified: 2019-08-21

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Copyright © 2019 Juniper Networks, Inc. All rights reserved.

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. and/or its affiliates in the United States and other countries. All other trademarks may be 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.

SRC PE Software CLI User Guide

Release 4.13.x

Copyright © 2019 Juniper Networks, Inc. All rights reserved.

Revision History

August 2019—Revision 1

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. 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

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement (“EULA”) posted at <https://support.juniper.net/support/eula/>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Abbreviated Table of Contents

	About the Documentation	xiii
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	21
Chapter 4	Getting Online Help for the SRC CLI	37
Part 2	Operational Mode and Configuration Mode	
Chapter 5	Using the SRC CLI Operational Commands to Monitor the SRC Software	47
Chapter 6	Using Commands and Statements to Configure the SRC Software	61
Chapter 7	Filtering Command Output in the SRC CLI	83
Chapter 8	Using Keyboard Shortcuts in the SRC CLI	95
Part 3	SRC CLI Environment and SRC Module and Component Management	
Chapter 9	Controlling the SRC CLI Environment	99
Chapter 10	Managing SRC Configurations	107
Chapter 11	Managing SRC Modules and Components with the CLI	131

Table of Contents

	About the Documentation	xiii
	SRC Documentation and Release Notes	xiii
	Audience	xiii
	Documentation Conventions	xiii
	Documentation Feedback	xv
	Requesting Technical Support	xvi
	Self-Help Online Tools and Resources	xvi
	Creating a Service Request with JTAC	xvi
Part 1	Introduction	
Chapter 1	Introducing the SRC CLI	3
	SRC CLI Overview	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	9
	Verifying System Status	10
	Configuring a User Account	12
	Creating a Configuration	14
	Using Shortcuts to Create a Configuration	17
	Making Changes to the Configuration	17
	Rolling Back Configuration Changes	18
Chapter 3	SRC CLI Basics	21
	Elements of the Command-Line Interface	21
	SRC CLI Messages	22
	Displaying Command Output	24
	Types of SRC Commands and Statements	25
	Command Options	26
	Configuration Statements and Identifiers	26

	Privilege Levels for Using Commands and Statements	28
	Switching Between Operational Mode and Configuration Mode	28
	Switching to Configuration Mode	28
	Returning to Operational Mode	29
	Running Operational Mode Commands from Configuration Mode	29
	Moving Among Hierarchy Levels in the SRC CLI	31
	Displaying SRC CLI Command History	32
	Configuring a Schedule for Executing the Commands or Scripts (SRC CLI)	32
Chapter 4	Getting Online Help for the SRC CLI	37
	Getting Help for Commands and Statements for the SRC CLI	37
	Getting Help for Omitted Statements	38
	Using SRC CLI Command Completion	39
	Using Command Completion in Configuration Mode	40
Part 2	Operational Mode and Configuration Mode	
Chapter 5	Using the SRC CLI Operational Commands to Monitor the SRC Software	47
	SRC CLI Command Categories	47
	Output Control Keys for monitor Command	48
	Commonly Used Operational Mode Commands	49
	Viewing Files and Directories	50
	Directories on the C Series Controller	50
	Listing Files and Directories	51
	Specifying Filenames and URLs	53
	Managing SRC Modules and Components	54
	Viewing C Series Controller Information	54
	Restarting an SRC Module or Component	56
	Stopping the SRC Software	57
	Rebooting the SRC Software	58
	Viewing Information about Users Logged Into the SRC Software	59
Chapter 6	Using Commands and Statements to Configure the SRC Software	61
	Understanding SRC CLI Configuration Mode	61
	Configuration Mode Commands	61
	Configuration Statements	63
	Configuration Statement Hierarchy	63
	Working in Configuration Mode with the SRC CLI	65
	Creating an SRC Configuration	65
	Entering Configuration Mode	65
	Exiting from Configuration Mode	66
	Modifying the Configuration	67
	Commands to Modify a Configuration	68
	Entering Values for Statement Options	68
	Displaying the Current Configuration	69
	Adding Configuration Statements and Identifiers	72
	Deleting a Statement from the Configuration	74
	Renaming an Identifier	76
	Inserting a New Identifier	77

	Copying a Configuration from One Configuration Location to Another	78
	Displaying Set Commands for the Configuration (SRC CLI)	79
	Verifying a Configuration	80
	Committing a Configuration	80
	Committing a Configuration and Exiting Configuration Mode	81
	When Multiple Users Configure the Software	82
Chapter 7	Filtering Command Output in the SRC CLI	83
	Using Keyboard Sequences at the MORE Prompt in the SRC CLI	83
	Using the Pipe () Symbol When Entering Commands	85
	Using Regular Expressions with the Pipe Symbol	86
	Using Pipe Filter Functions	87
	Counting the Number of Lines of Output	87
	Displaying Summarized Output for a Hierarchy Level (SRC CLI)	87
	Displaying Output in XML Tag Format	88
	Disregarding Output That Does Not Match a Regular Expression	89
	Displaying Output from the First Match of a Regular Expression	89
	Displaying the End of the Output for a Command	90
	Displaying Output That Matches a Regular Expression	91
	Preventing Output from Being Paginated	92
	Saving Output to a File	92
Chapter 8	Using Keyboard Shortcuts in the SRC CLI	95
Part 3	SRC CLI Environment and SRC Module and Component Management	
Chapter 9	Controlling the SRC CLI Environment	99
	Commands to Control the SRC CLI Environment Overview	99
	Setting the Editing Level for the SRC CLI	100
	Setting the Terminal Type for the SRC CLI	101
	Setting the Language for the Terminal Environment for the SRC CLI	101
	Setting the Screen Length for the SRC CLI	102
	Setting the Screen Width for the SRC CLI	102
	Changing the Password for the SRC CLI	103
	Setting the SRC CLI Prompt	103
	Setting the Directory for the SRC CLI	104
	Setting Command Completion for the SRC CLI	104
	Viewing Settings for the SRC CLI	105
Chapter 10	Managing SRC Configurations	107
	How the SRC Configuration Is Stored	107
	Commands to Save a Configuration to a File	108
	SSH Host Keys	109
	Component State	109
	Updating the SRC Configuration	109
	SRC Configuration Updates	109
	Before You Load a Configuration	110
	Commands to Load a Configuration	110
	SSH Host Keys	112
	Component State	112

	Attributes in SRC Configuration Files	113
	About SRC Configuration Files in XML Format	113
	Example: Using Attributes When Editing an XML Configuration File	115
	About SRC Configuration Files in Text Format	116
	Example: Using Attributes When Editing a Text Configuration File	117
	About SRC Configuration Files in Set Format	118
	Preparing a File to Be Loaded into the Current SRC Configuration	119
	Loading an SRC Configuration	120
	Replacing the Current Configuration with the Default SRC Configuration . .	120
	Merging the Active Configuration with Another Configuration	121
	Replacing the Configuration	123
	Replacing Parts of the Configuration	124
	Adding a Configuration Through Configuration Mode Commands	126
	Loading a Configuration at a Specified Hierarchy Level	127
	Comparing SRC Configurations	127
	Reverting to a Previous SRC Configuration	128
	Cutting and Pasting Configuration Information at the SRC CLI	129
Chapter 11	Managing SRC Modules and Components with the CLI	131
	Verifying Status of SRC Components	131
	Enabling SRC Components	132
	Disabling an SRC Component	133
	Restarting an SRC Component	134

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	21
	Figure 4: Elements of the Command-Line Interface	21
	Figure 5: Command Prompt in Configuration Mode	22
	Figure 6: Hierarchy-Level Banner	22
	Figure 7: The MORE Prompt	24
	Figure 8: Command Options	26
Part 2	Operational Mode and Configuration Mode	
Chapter 6	Using Commands and Statements to Configure the SRC Software	61
	Figure 9: Sample Configuration Mode Hierarchy of Statements	64
Part 3	SRC CLI Environment and SRC Module and Component Management	
Chapter 10	Managing SRC Configurations	107
	Figure 10: Commands for Storing and Modifying the Configuration	108

List of Tables

	About the Documentation	xiii
	Table 1: Notice Icons	xiv
	Table 2: Text Conventions	xiv
Part 1	Introduction	
Chapter 3	SRC CLI Basics	21
	Table 3: MORE Prompt Keyboard Sequences	24
	Table 4: CLI Configuration Mode Navigation Commands	31
	Table 5: Values Allowed for the Periodic Strings	34
	Table 6: Available Special Characters	34
	Table 7: Special String Options	35
Part 2	Operational Mode and Configuration Mode	
Chapter 5	Using the SRC CLI Operational Commands to Monitor the SRC Software	47
	Table 8: Output Control Keys for the monitor Command	48
	Table 9: Commonly Used Operational Mode Commands	49
	Table 10: Directories on a C Series Controller	50
	Table 11: Options to Restart an SRC Module or Component	56
Chapter 6	Using Commands and Statements to Configure the SRC Software	61
	Table 12: Summary of Configuration Mode Commands	61
	Table 13: Configuration Mode Top-Level Statements	63
Chapter 7	Filtering Command Output in the SRC CLI	83
	Table 14: MORE Prompt Keyboard Sequences	83
	Table 15: Common Regular Expression Operators in Operational Mode Commands	86
Chapter 8	Using Keyboard Shortcuts in the SRC CLI	95
	Table 16: CLI Keyboard Sequences	95
Part 3	SRC CLI Environment and SRC Module and Component Management	
Chapter 9	Controlling the SRC CLI Environment	99
	Table 17: Editing Levels	100

About the Documentation

- SRC Documentation and Release Notes on page xiii
- Audience on page xiii
- Documentation Conventions on page xiii
- Documentation Feedback on page xv
- Requesting Technical Support on page xvi

SRC Documentation and Release Notes

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

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 xiv](#) defines the notice icons used in this guide. [Table 2 on page xiv](#) 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.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

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# ... https://www.juniper.net/documentation/software/management/src/api-index.html

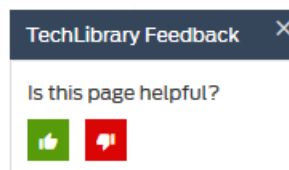
Table 2: Text Conventions (continued)

<i>Italic sans serif typeface</i>	Represents variables in SRC CLI commands.	<code>user@host# set local-address local-address</code>
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <gfwif>.
Key name	Indicates the name of a key on the keyboard.	Press Enter.
Key names linked with a plus sign (+)	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
<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.	<code>Plugin.radiusAcct-1.class=\ net.juniper.smgmt.sae.plugin\ RadiusTrackingPluginEvent</code>
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.)	<code>diagnostic line</code>

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the [Juniper Networks TechLibrary](#) site, and do one of the following:



- Click the thumbs-up icon if the information on the page was helpful to you.
- Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software 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 Partner Support Service 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 <https://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: <https://www.juniper.net/customers/support/>
- Search for known bugs: <https://prsearch.juniper.net/>
- Find product documentation: <https://www.juniper.net/documentation/>
- Find solutions and answer questions using our Knowledge Base: <https://kb.juniper.net/>
- Download the latest versions of software and review release notes: <https://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum: <https://www.juniper.net/company/communities/>
- Create a service request online: <https://myjuniper.juniper.net>

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

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit <https://myjuniper.juniper.net>.
- 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 <https://support.juniper.net/support/requesting-support/>.

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 21](#)
- [Getting Online Help for the SRC CLI on page 37](#)

CHAPTER 1

Introducing the SRC CLI

- [SRC CLI Overview 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](#)

SRC CLI Overview

The SRC CLI is a Junos OS–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 command 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 21](#)
- [*Configuring Access to the SRC CLI Overview*](#)
- [Types of SRC Commands and Statements on page 25](#)
- [Understanding SRC CLI Command Modes on page 3](#)
- [Understanding SRC Command and Statement Hierarchies on page 4](#)

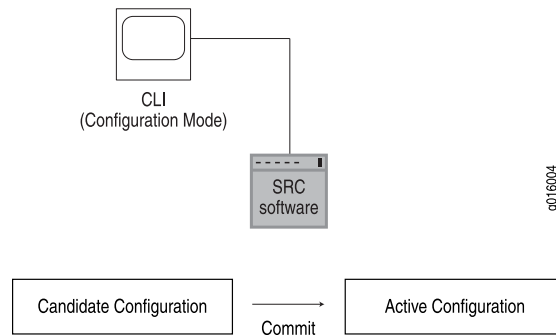
Understanding SRC CLI Command Modes

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 47](#)
- [Commonly Used Operational Mode Commands on page 49](#)
- [Understanding SRC CLI Configuration Mode on page 61](#)
- [Understanding SRC Command and Statement Hierarchies on page 4](#)
- [Key Features of the SRC CLI on page 5](#)

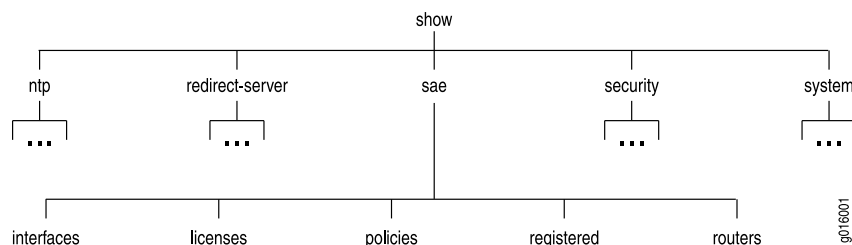
Understanding SRC Command and Statement Hierarchies

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



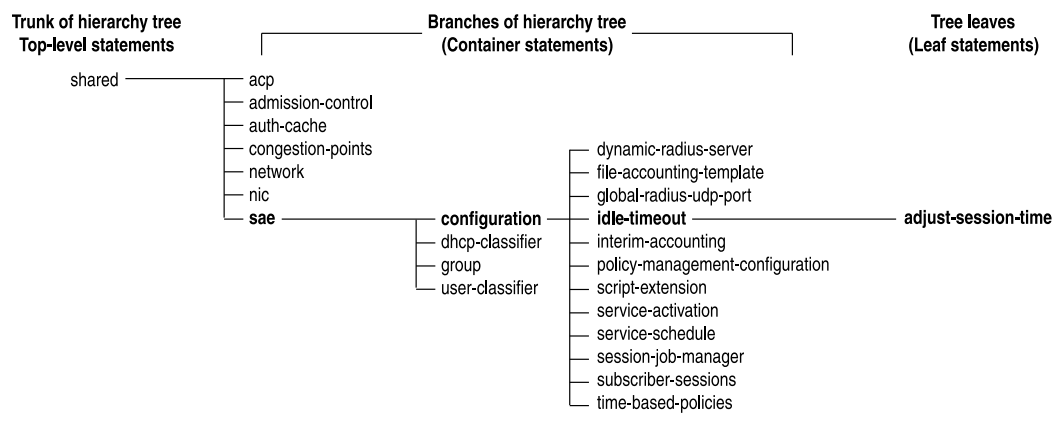
To execute a command, you enter the full command name, starting at the top level of the hierarchy. For example, to display information about SAE licenses, you enter the **show sae licenses** command.

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

- [Commands to Control the SRC CLI Environment Overview on page 99](#)
- [Types of SRC Commands and Statements on page 25](#)
- [SRC CLI Command Categories on page 47](#)
- [Moving Among Hierarchy Levels in the SRC CLI on page 31](#)
- [Getting Help for Commands and Statements for the SRC CLI on page 37](#)

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 OS, 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

- [SRC CLI Overview on page 3](#)
- [Using SRC CLI Command Completion on page 39](#)
- [Using Keyboard Sequences at the MORE Prompt in the SRC CLI on page 83](#)
- [Getting Help for Commands and Statements for the SRC CLI on page 37](#)

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 9](#)
- [Verifying System Status on page 10](#)
- [Configuring a User Account on page 12](#)
- [Creating a Configuration on page 14](#)
- [Using Shortcuts to Create a Configuration on page 17](#)
- [Making Changes to the Configuration on page 17](#)
- [Rolling Back Configuration Changes on page 18](#)

Before You Start the SRC CLI

Make sure the SRC software has been configured for remote access through SSH. To enable root access over an SSH connection, you must configure the **system services ssh root-login allow** statement.



NOTE: For security reasons, we recommend that you do not allow remote users to access the CLI as root. Instead, you can use the following statement to create a username and password that a user can use to log in to the C Series Controller as root through SSH:

```
system login user user-name {  
  class class;  
  authentication {  
    plain-text-password;  
    encrypted-password " password ";  
    ssh-authorized-keys [ssh-authorized-keys ...];  
  }  
}
```

For information about initial CLI configuration, see the *SRC PE Getting Started Guide*.

- Related Documentation**
- [Starting the SRC CLI on page 8](#)
 - [Key Features of the SRC CLI on page 5](#)
 - [SRC CLI Overview on page 3](#)
 - [Configuring User Accounts \(SRC CLI\)](#)

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 #.

For information about the SRC CLI, see [“SRC CLI Overview” on page 3](#).

- Related Documentation**
- [Before You Start the SRC CLI on page 7](#)
 - [Displaying Commands on page 9](#)
 - [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](#)

Displaying Commands

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

To use the various Help commands:

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

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

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

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

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

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

- Related Documentation**
- [Getting Help for Commands and Statements for the SRC CLI on page 37](#)
 - [Commands to Control the SRC CLI Environment Overview on page 99](#)
 - [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
  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 the status of installed components.

```
user@host> show component
Installed Components
Name      Version      Status
---      -
acp       Release: 7.8 Build: ACP.A.MAIN.1480      disabled
activity  Release: 7.8 Build: ACTIVITY.A.MAIN.1480  running
agent     Release: 7.8 Build: SYSMAN.A.MAIN.1480    disabled
appsvr    Release: 7.8 Build: JBOSS.A.MAIN.1480     disabled
cli       Release: MAIN Build: CLI.A.MAIN.1480       running
database  Release: 7.8 Build: SSR.A.MAIN.1480       disabled
diameter  Release: 7.8 Build: DIAMETER.A.MAIN.1480  running
dsa       Release: 7.8 Build: GATEWAYAPPS.A.MAIN.1480 disabled
editor    Release: 7.8 Build: EDITOR.A.MAIN.1480    running
extsubmon Release: 7.8 Build: MONAGENT.A.MAIN.1480   disabled
gw-3gpp   Release: 7.8 Build: 3GPPGW.A.MAIN.1480    disabled
gy-3gpp   Release: 7.8 Build: 3GPPGY.A.MAIN.1480    running
ims       Release: 7.8 Build: IMS.A.MAIN.1480       disabled
jdb       Release: 7.8 Build: DIRXA.A.MAIN.1480     running
licSvr    Release: 7.8 Build: LICSVR.A.MAIN.1480    disabled
naming    Release: 7.8 Build: NAMING.A.MAIN.1480     running
```

nic	Release: 7.8 Build: GATEWAY.A.MAIN.1480	running
redir	Release: 7.8 Build: REDIR.A.MAIN.1480	disabled
sae	Release: 7.8 Build: SAE.A.MAIN.1480	running
sic	Release: 7.8 Build: SICCLI.A.MAIN.1480	disabled
vta	Release: 7.8 Build: VTA.A.MAIN.1480	disabled
webadm	Release: 7.8 Build: WEBADM.A.MAIN.1480	disabled

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



NOTE: When you issue the **show system information** command in a virtualized SRC software, the manufacturer, version, and serial number details are not displayed in the output. In addition, the product name is displayed as vSRC.

```
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 Core(s) 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          30.000 degrees C
Physical CPU-1  34.000 degrees C
Physical CPU-2  29.000 degrees C
```

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

- Related Documentation**
- [Viewing C Series Controller Information on page 54](#)
 - [Viewing Settings for the SRC CLI on page 105](#)
 - *Viewing Information About the SRC CLI (C-Web Interface)*
 - *Configuring Access to the SRC CLI Overview*

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 17](#)
- [Making Changes to the Configuration on page 17](#)
- [Initially Configuring the SAE](#)
- [Displaying the Current Configuration on page 69](#)
- [Verifying a Configuration on page 80](#)

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 14](#)
- [Committing a Configuration on page 80](#)
- [Displaying the Current Configuration on page 69](#)
- [Verifying a Configuration on page 80](#)

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 18](#)
- [Committing a Configuration on page 80](#)
- [Displaying the Current Configuration on page 69](#)
- [Verifying a Configuration on page 80](#)
- [Commands to Modify a Configuration on page 68](#)

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 17](#)
- [Displaying the Current Configuration on page 69](#)
- [Verifying a Configuration on page 80](#)
- [Commands to Modify a Configuration on page 68](#)

CHAPTER 3

SRC CLI Basics

- Elements of the Command-Line Interface on page 21
- SRC CLI Messages on page 22
- Displaying Command Output on page 24
- Types of SRC Commands and Statements on page 25
- Switching Between Operational Mode and Configuration Mode on page 28
- Moving Among Hierarchy Levels in the SRC CLI on page 31
- Displaying SRC CLI Command History on page 32
- Configuring a Schedule for Executing the Commands or Scripts (SRC CLI) on page 32

Elements of the Command-Line Interface

Figure 4 on page 21 shows elements of the command-line interface in operational mode.

Figure 4: Elements of the Command-Line Interface

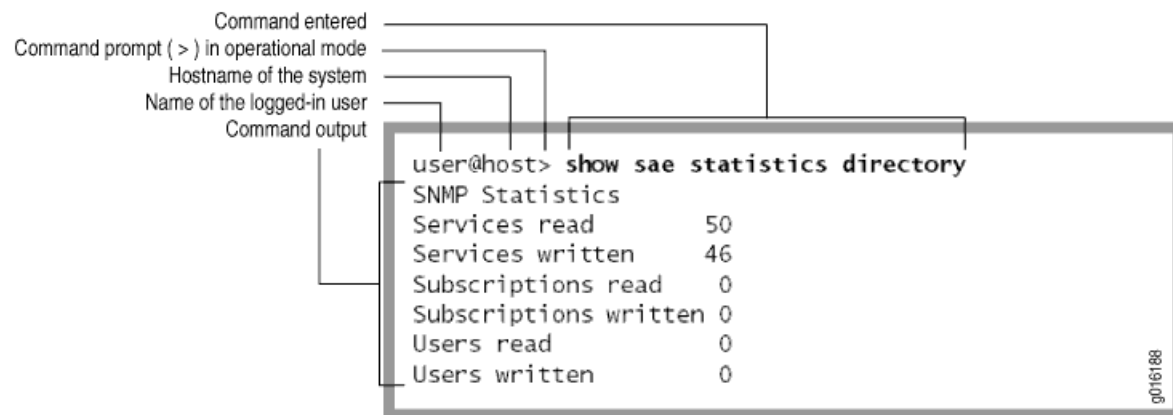
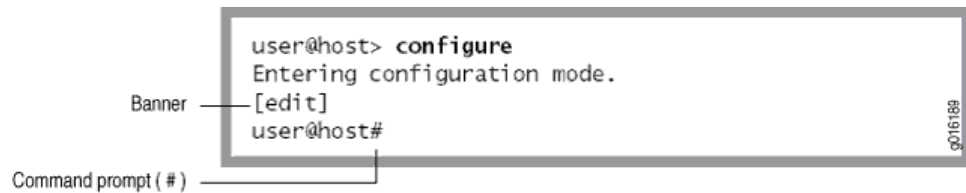


Figure 5 on page 22 shows elements of the command-line interface in configuration mode. In configuration mode, the prompt changes from a > to a #.

Figure 5: Command Prompt in Configuration Mode



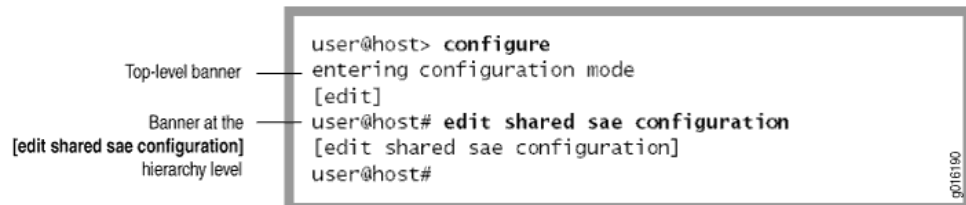
```

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 22](#).)

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

- [SRC CLI Overview 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 105](#)
- [Configuring Access to the SRC CLI Overview](#)

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 31](#)
- [Using Command Completion in Configuration Mode on page 40](#)
- [Entering Configuration Mode on page 65](#)
- [Exiting from Configuration Mode on page 66](#)
- [Setting Command Completion for the SRC CLI on page 104](#)

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 24](#).)

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 ——— -- MORE --

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 24](#) 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

Table 3: MORE Prompt Keyboard Sequences (continued)

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

Related Documentation

- [Displaying the End of the Output for a Command on page 90](#)
- [Displaying Output That Matches a Regular Expression on page 91](#)
- [Saving Output to a File on page 92](#)
- [Setting the Screen Length for the SRC CLI on page 102](#)
- [Using Keyboard Sequences at the MORE Prompt in the SRC CLI on page 83](#)

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 47](#).

- 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 [“Commands to Control the SRC CLI Environment Overview” on page 99](#).

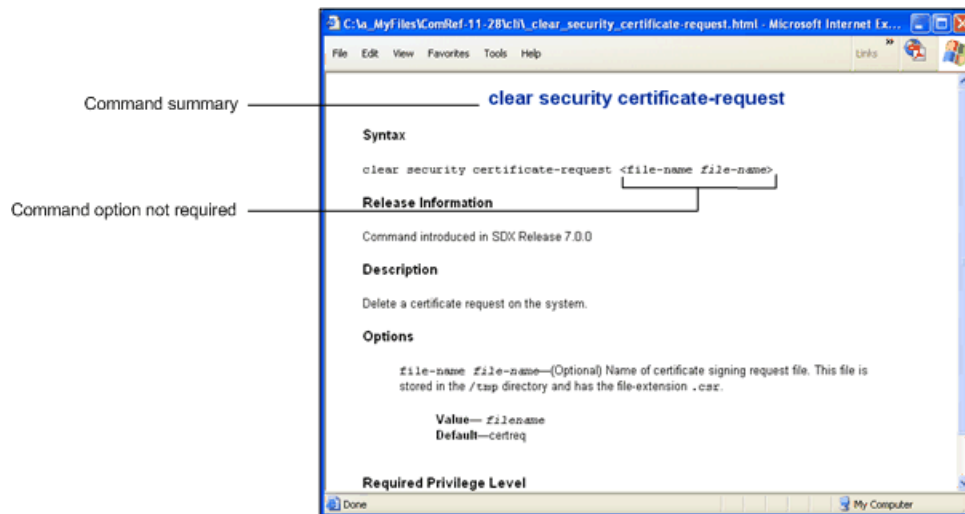
- Configuration mode commands—Commands that you enter in configuration mode are used to perform general configuration functions; for example, committing a configuration, navigating the hierarchy, and managing configuration files. For more information, see [“Understanding SRC CLI Configuration Mode” on page 61](#).
- 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 Reference* show which options are required and which options are not. Options at the top statement level that are not required are shown with angle brackets (<>). (See [Figure 8 on page 26](#).) 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
  disable-monitor      NTP disable monitor configuration
> 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 **ssh** command. 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 [“Commands to Control the SRC CLI Environment Overview” on page 99](#).

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 49](#)
- [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 28](#)
- [Returning to Operational Mode on page 29](#)
- [Running Operational Mode Commands from Configuration Mode on page 29](#)

Switching to Configuration Mode

To enter configuration mode:

- Type the **configure** command or the **edit** command from the CLI operation mode. For example:

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

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

Returning to Operational Mode

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

To commit the configuration and exit:

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

To exit without committing:

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

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

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

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

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

user@host>
```

Running Operational Mode Commands from Configuration Mode

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

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

[edit]

user@host# *run operational-mode-command*

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

```
[edit shared sae configuration]
user@host# show
plug-ins {
  pool rksPlugin {
    pcmm-rks {
      load-balancing-mode failover;
      failback-timer -1;
      retry-interval 3000;
      maximum-queue-length 10000;
      feid-mso-domain-name abcd.com;
      trusted-element;
      default-peer radius01;
      peer-group rksPeer {
        server-address 10.10.3.60;
        server-port 1812;
      }
    }
  }
}
driver {
  . . .
```

[edit shared sae configuration]

user@host> **run show component**

Installed Components

Name	Version	Status
acp	Release: 7.8 Build: ACP.A.MAIN.1480	disabled
activity	Release: 7.8 Build: ACTIVITY.A.MAIN.1480	running
agent	Release: 7.8 Build: SYSMAN.A.MAIN.1480	disabled
appsvr	Release: 7.8 Build: JBOSS.A.MAIN.1480	disabled
cli	Release: MAIN Build: CLI.A.MAIN.1480	running
database	Release: 7.8 Build: SSR.A.MAIN.1480	disabled
diameter	Release: 7.8 Build: DIAMETER.A.MAIN.1480	running
dsa	Release: 7.8 Build: GATEWAYAPPS.A.MAIN.1480	disabled
editor	Release: 7.8 Build: EDITOR.A.MAIN.1480	running
extsubmon	Release: 7.8 Build: MONAGENT.A.MAIN.1480	disabled
gw-3gpp	Release: 7.8 Build: 3GPPGW.A.MAIN.1480	disabled
gy-3gpp	Release: 7.8 Build: 3GPPGY.A.MAIN.1480	running
ims	Release: 7.8 Build: IMS.A.MAIN.1480	disabled
jdb	Release: 7.8 Build: DIRXA.A.MAIN.1480	running
licSvr	Release: 7.8 Build: LICSVR.A.MAIN.1480	disabled
naming	Release: 7.8 Build: NAMING.A.MAIN.1480	running
nic	Release: 7.8 Build: GATEWAY.A.MAIN.1480	running
redir	Release: 7.8 Build: REDIR.A.MAIN.1480	disabled
sae	Release: 7.8 Build: SAE.A.MAIN.1480	running
sic	Release: 7.8 Build: SICCLI.A.MAIN.1480	disabled

vta	Release: 7.8 Build: VTA.A.MAIN.1480	disabled
webadm	Release: 7.8 Build: WEBADM.A.MAIN.1480	disabled

Related Documentation

- [Entering Configuration Mode on page 65](#)
- [Exiting from Configuration Mode on page 66](#)
- [Displaying SRC CLI Command History on page 32](#)
- [Commonly Used Operational Mode Commands on page 49](#)
- [Types of SRC Commands and Statements on page 25](#)

Moving Among Hierarchy Levels in the SRC CLI

You can use the CLI commands in [Table 4 on page 31](#) 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 65](#)
- [Entering Configuration Mode on page 65](#)
- [Using Command Completion in Configuration Mode on page 40](#)
- [Understanding SRC Command and Statement Hierarchies on page 4](#)
- [Understanding SRC CLI Configuration Mode on page 61](#)

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 9](#)
- [Getting Help for Commands and Statements for the SRC CLI on page 37](#)
- [Types of SRC Commands and Statements on page 25](#)
- [Understanding SRC Command and Statement Hierarchies on page 4](#)
- [SRC CLI Command Categories on page 47](#)

Configuring a Schedule for Executing the Commands or Scripts (SRC CLI)

To periodically execute the SRC CLI commands or scripts according to a given schedule, use the **system schedule *schedule-name*** command. You can use the redirection operator (>) with the commands or scripts to redirect the command execution output to a file.

Use the following statements to schedule a repetitive task on an SRC system:

```
system schedule schedule-name {
    day-of-week day-of-week;
    month month;
    day-of-month day-of-month;
    hour hour;
    minute minute;
```



```
special (reboot | yearly | annually | monthly | weekly | daily | midnight | hourly);  
command command;  
script script;  
}
```

To periodically execute the commands or scripts according to a given schedule:

1. In configuration mode, enter the name of the schedule that is capable of adding schedule entries for the executable scripts or commands. A schedule name can contain alphanumeric characters only. In this sample procedure, **src1cron** is the name of the schedule.

```
[edit]  
user@host# edit system schedule src1cron
```

2. (Optional) Enter the day of the week on which you want the command or script sequence to execute. The default value is * (asterisk).

```
[edit system schedule src1cron]  
user@host# set day-of-week day-of-week
```

3. (Optional) Enter the month of the year in which you want the command or script sequence to execute. The default value is * (asterisk).

```
[edit system schedule src1cron]  
user@host# set month month
```

4. (Optional) Enter the day of the month on which you want the command or script sequence to execute. The default value is * (asterisk).

```
[edit system schedule src1cron]  
user@host# set day-of-month day-of-month
```

5. (Optional) Enter the hour of the day at which you want the command or script sequence to execute. The default value is * (asterisk).

```
[edit system schedule src1cron]  
user@host# set hour hour
```

6. (Optional) Enter the minute at which you want the command or script sequence to execute. The default value is * (asterisk).

```
[edit system schedule src1cron]
user@host# set minute minute
```

[Table 5 on page 34](#) lists the values allowed for the periodic strings (such as **day-of-week**, **month**, **day-of-month**, **hour**, and **minute**).

Table 5: Values Allowed for the Periodic Strings

Options	Values
day-of-week	0–7 (where both 0 and 7 mean Sunday) or SUN-SAT
month	1–12 or JAN-DEC
day-of-month	1–31
hour	0–23
minute	0–59

You can use multiple combinations of the special characters described in [Table 6 on page 34](#).

Table 6: Available Special Characters

Special Characters	Meaning
*	Indicates all values. For example, to execute the command or script every hour: user@host# set hour *
/	Indicates the increments of ranges. For example, to execute the command or script at the third minute of the hour and every 15 minutes thereafter: user@host# set minute 3-59/15
,	Indicates to separate the values of a list. For example, to execute the command or script on Mondays, Wednesdays and Fridays: user@host# day of week MON,WED,FRI
-	Indicates the ranges. For example, to execute the command or script every hour from 3 through 10: user@host# set hour 3-10

7. (Optional) Enter the special string values such as **reboot**, **yearly**, **annually**, **monthly**, **weekly**, **daily**, **midnight**, and **hourly** at which you want the command or script sequence to execute.

For example, to execute the command or script at midnight on the first day of each month:

```
[edit system schedule src/cron]
user@host# set special-string monthly
```



NOTE: If you configure both special string values and periodic string values (such as **day-of-week**, **month**, **day-of-month**, **hour**, or **minute** options), a message indicating that both special strings and periodic strings cannot be defined is displayed when you commit the changes.

Table 7 on page 35 lists the special string options.

Table 7: Special String Options

Options	Description
reboot	Executes the command or script at boot and reboot of the system.
yearly	Executes the command or script at midnight, January 1 each year.
annually	Executes the command or script at midnight, January 1 each year.
monthly	Executes the command or script at midnight on the first day of each month.
weekly	Executes the command or script at midnight each Sunday.
daily	Executes the command or script at midnight each day.
midnight	Executes the command or script at midnight each day.
hourly	Executes the command or script at on the first second of every hour.

8. Configure the complete path of the script to be executed for the schedule.



NOTE: The **script** and **command** options are mutually exclusive. You cannot configure both options at the same time.

```
[edit system schedule src/cron]
user@host# set script script
```



NOTE: The CLI editing level must be set to expert by using the `set cli level expert` command for this option.

9. Configure the CLI command to be executed for the schedule.



NOTE:

- The script and command options are mutually exclusive. You cannot configure both options at the same time.

Before you configure a CLI command for the schedule, perform the following steps to make sure that you can configure the CLI command as part of the schedule execution:

1. Enter the CLI command in the shell with the `cli -c "command name"` option.
2. Verify whether the CLI command returns the expected output.

```
[edit system schedule src1cron]
user@host# set command command
```

10. (Optional) Verify your configuration.

```
user@host# show
src1cron {
  command show;
  day-of-month 7;
  day-of-week 5;
  hour 11;
  minute 0;
  month 12;
}
```

- Related Documentation**
- [SRC Script Services Overview](#)
 - [SRC CLI Overview on page 3](#)

CHAPTER 4

Getting Online Help for the SRC CLI

- [Getting Help for Commands and Statements for the SRC CLI on page 37](#)
- [Getting Help for Omitted Statements on page 38](#)
- [Using SRC CLI Command Completion on page 39](#)
- [Using Command Completion in Configuration Mode on page 40](#)

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 38](#)
- [Commands to Control the SRC CLI Environment Overview on page 99](#)
- [Understanding SRC Command and Statement Hierarchies on page 4](#)
- [Types of SRC Commands and Statements on page 25](#)
- [SRC CLI Messages on page 22](#)

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 37](#)
- [Types of SRC Commands and Statements on page 25](#)
- [Adding Configuration Statements and Identifiers on page 72](#)
- [SRC CLI Messages on page 22](#)

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 37](#)
- [Using Command Completion in Configuration Mode on page 40](#)
- [Setting Command Completion for the SRC CLI on page 104](#)
- [SRC CLI Command Categories on page 47](#)
- [SRC CLI Messages on page 22](#)

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:

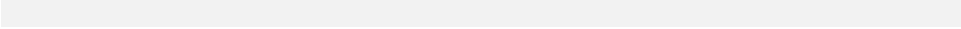
```
[edit]
user@host# edit system l?
Possible completions:
  > ldap         LDAP properties
  > login        Login properties
user@host# edit system l
```

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 39](#)
 - [Getting Help for Commands and Statements for the SRC CLI on page 37](#)
 - [Setting Command Completion for the SRC CLI on page 104](#)
 - [Understanding SRC CLI Configuration Mode on page 61](#)

PART 2

Operational Mode and Configuration Mode

- [Using the SRC CLI Operational Commands to Monitor the SRC Software on page 47](#)
- [Using Commands and Statements to Configure the SRC Software on page 61](#)
- [Filtering Command Output in the SRC CLI on page 83](#)
- [Using Keyboard Shortcuts in the SRC CLI on page 95](#)

CHAPTER 5

Using the SRC CLI Operational Commands to Monitor the SRC Software

- [SRC CLI Command Categories on page 47](#)
- [Output Control Keys for monitor Command on page 48](#)
- [Commonly Used Operational Mode Commands on page 49](#)
- [Viewing Files and Directories on page 50](#)
- [Managing SRC Modules and Components on page 54](#)
- [Viewing Information about Users Logged Into the SRC Software on page 59](#)

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 25](#)
- [Displaying Commands on page 9](#)
- [Getting Help for Commands and Statements for the SRC CLI on page 37](#)
- [Commonly Used Operational Mode Commands on page 49](#)
- [Commands to Control the SRC CLI Environment Overview on page 99](#)

Output Control Keys for monitor Command

The output of the monitor command shows how much each field has changed since you started the command or since you cleared the counters by using the c key. For a description of the statistical information provided in the output of this command, see the show command output.

To control the output of the monitor command while it is running, use the keys listed in [Table 8 on page 48](#). The keys are not case-sensitive.

Table 8: Output Control Keys for the monitor Command

Key	Action
c	Clears (returns to zero) the delta counters since the command was started.
f	Freezes the display, halting the display of updated statistics and delta counters.
q or Esc	Quits the command and returns to the command prompt.
t	Thaws the display, resuming the update of the statistics and delta counters.

Table 8: Output Control Keys for the monitor Command (continued)

Key	Action
a	Displays the real-time statistics of the next action. NOTE: This key is applicable only for the VTA monitor commands.
e	Displays the real-time statistics of the next event handler. NOTE: This key is applicable only for the VTA monitor commands.

Related Documentation

- *SRC CLI Commands to Monitor Dynamic Service Activator*
- *Monitoring Statistics for External Subscriber Monitor (SRC CLI)*
- *Monitoring SRC VTA Performance Statistics (SRC CLI)*
- *Monitoring Statistics for Subscriber and Service Sessions (SRC CLI)*

Commonly Used Operational Mode Commands

Table 9 on page 49 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 9: 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
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 25](#)
 - [SRC CLI Command Categories on page 47](#)
 - [Commands to Control the SRC CLI Environment Overview on page 99](#)
 - [Viewing C Series Controller Information on page 54](#)

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 50](#)
- [Listing Files and Directories on page 51](#)
- [Specifying Filenames and URLs on page 53](#)

Directories on the C Series Controller

The C Series Controller has numerous directories used by the operating system. [Table 10 on page 50](#) lists directories on a C Series Controller.

Table 10: Directories on a C Series Controller

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

- See Also**
- [Listing Files and Directories on page 51](#)
 - [Specifying Filenames and URLs on page 53](#)
 - [Viewing C Series Controller Information on page 54](#)

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
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 mktmp-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.
. . .
```

- See Also**
- [Setting the Directory for the SRC CLI on page 104](#)
 - [Specifying Filenames and URLs on page 53](#)
 - [Directories on the C Series Controller on page 50](#)
 - [Commands to Control the SRC CLI Environment Overview on page 99](#)
 - [Commonly Used Operational Mode Commands on page 49](#)

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 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.



NOTE: This format supports only a single partition on a USB storage device. If the system device of the USB storage device contains more than one partition, the SRC software sorts the partitions according to file system type and uses the first partition.

- **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.
```

- See Also**
- [Listing Files and Directories on page 51](#)
 - [Directories on the C Series Controller on page 50](#)
 - [Commonly Used Operational Mode Commands on page 49](#)
 - [Recovering or Installing System Software on a C Series Controller by Using the USB Storage Device Supplied by Juniper Networks](#)

Managing SRC Modules and Components

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

- [Viewing C Series Controller Information on page 54](#)
- [Restarting an SRC Module or Component on page 56](#)
- [Stopping the SRC Software on page 57](#)
- [Rebooting the SRC Software on page 58](#)

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.



NOTE: When you issue the `show system information` command in a virtualized SRC software, the manufacturer, version, and serial number details are not displayed in the output. In addition, the product name is displayed as vSRC.

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 CPU Core(s) 4
CPU ModelDual Core AMD Opteron(tm) Processor 265
Clock Speed1804.158 MHz
```

```
Disk Information
Mountpoint  Total Used Use%
/ 2015M 1310M 64%
/altroot2015M 35M 1%
/altvar29G 75M 0%
/boot 98M 14M 14%
/var 31G 1850M 5%
```

```
Temperature
System 30.000 degrees C
Physical CPU-1 34.000 degrees C
Physical CPU-2 29.000 degrees 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
acp       Release: 7.8 Build: ACP.A.MAIN.1480      disabled
activity  Release: 7.8 Build: ACTIVITY.A.MAIN.1480  running
agent     Release: 7.8 Build: SYSMAN.A.MAIN.1480    disabled
appsvr    Release: 7.8 Build: JBOSS.A.MAIN.1480     disabled
cli       Release: MAIN Build: CLI.A.MAIN.1480      running
database  Release: 7.8 Build: SSR.A.MAIN.1480       disabled
diameter  Release: 7.8 Build: DIAMETER.A.MAIN.1480  running
dsa       Release: 7.8 Build: GATEWAYAPPS.A.MAIN.1480 disabled
editor    Release: 7.8 Build: EDITOR.A.MAIN.1480    running
extsubmon Release: 7.8 Build: MONAGENT.A.MAIN.1480   disabled
gw-3gpp   Release: 7.8 Build: 3GPPGW.A.MAIN.1480    disabled
gy-3gpp   Release: 7.8 Build: 3GPPGY.A.MAIN.1480    running
ims       Release: 7.8 Build: IMS.A.MAIN.1480       disabled
jdb       Release: 7.8 Build: DIRXA.A.MAIN.1480     running
licSvr    Release: 7.8 Build: LICSVR.A.MAIN.1480    disabled
naming    Release: 7.8 Build: NAMING.A.MAIN.1480    running
nic       Release: 7.8 Build: GATEWAY.A.MAIN.1480   running
redir     Release: 7.8 Build: REDIR.A.MAIN.1480     disabled
sae       Release: 7.8 Build: SAE.A.MAIN.1480       running
sic       Release: 7.8 Build: SICCLI.A.MAIN.1480    disabled
vta       Release: 7.8 Build: VTA.A.MAIN.1480       disabled
webadm    Release: 7.8 Build: WEBADM.A.MAIN.1480    disabled
```

- See Also**
- [Directories on the C Series Controller on page 50](#)
 - [Verifying Status of SRC Components on page 131](#)

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 11 on page 56](#) shows options available for the **restart** command.

Table 11: Options to Restart an SRC Module or Component

Option	Description
<i>component</i>	Name of the module or component to restart.

Table 11: Options to Restart an SRC Module or Component (continued)

Option	Description
gracefully component	Restart a specified module or component by sending the equivalent of a UNIX SIGTERM signal.
immediately component	Immediately restart a module or component by sending the equivalent of a UNIX SIGKILL signal.
soft component	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...
```

- See Also**
- [Enabling SRC Components on page 132](#)
 - [Disabling an SRC Component on page 133](#)
 - [Stopping the SRC Software on page 57](#)
 - [Rebooting the SRC Software on page 58](#)

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
Broadcasting 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
```

- See Also**
- [Restarting an SRC Module or Component on page 56](#)
 - [Rebooting the SRC Software on page 58](#)

Rebooting the SRC Software

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

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

```
user@host>
request system reboot

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

- See Also**
- [Stopping the SRC Software on page 57](#)
 - [Restarting an SRC Module or Component on page 56](#)

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
```

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 82](#)
 - *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 61](#)
- [Working in Configuration Mode with the SRC CLI on page 65](#)
- [Modifying the Configuration on page 67](#)
- [Verifying a Configuration on page 80](#)
- [Committing a Configuration on page 80](#)
- [Committing a Configuration and Exiting Configuration Mode on page 81](#)
- [When Multiple Users Configure the Software on page 82](#)

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 12 on page 61](#) summarizes each CLI configuration mode command. The commands are listed alphabetically.

Table 12: Summary of Configuration Mode Commands

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

Table 12: 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, XML, or set 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 13 on page 63 describes top-level CLI configuration mode statements.

Table 13: 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*.

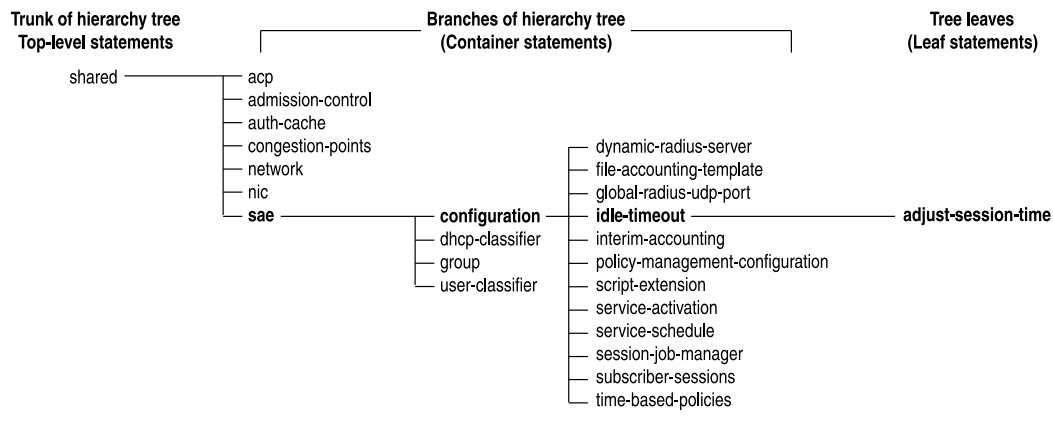
Configuration Statement Hierarchy

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 64 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 64 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 64 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 120](#)
- [Merging the Active Configuration with Another Configuration on page 121](#)
- [Replacing the Configuration on page 123](#)
- [Replacing Parts of the Configuration on page 124](#)
- [Adding a Configuration Through Configuration Mode Commands on page 126](#)
- [Loading a Configuration at a Specified Hierarchy Level on page 127](#)

Working in Configuration Mode with the SRC CLI

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

- [Creating an SRC Configuration on page 65](#)
- [Entering Configuration Mode on page 65](#)
- [Exiting from Configuration Mode on page 66](#)

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.

See Also

- [SRC Configuration Updates on page 109](#)
- [Understanding SRC CLI Configuration Mode on page 61](#)
- [Entering Configuration Mode on page 65](#)
- [Committing a Configuration and Exiting Configuration Mode on page 81](#)
- [When Multiple Users Configure the Software on page 82](#)
- [Verifying a Configuration on page 80](#)

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.
```

- See Also**
- [Exiting from Configuration Mode on page 66](#)
 - [Understanding SRC CLI Configuration Mode on page 61](#)
 - [Commands to Modify a Configuration on page 68](#)
 - [Creating an SRC Configuration on page 65](#)

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>
```

- See Also**
- [Entering Configuration Mode on page 65](#)
 - [Committing a Configuration and Exiting Configuration Mode on page 81](#)
 - [Understanding SRC CLI Configuration Mode on page 61](#)

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. Topics include:

- [Commands to Modify a Configuration on page 68](#)
- [Entering Values for Statement Options on page 68](#)
- [Displaying the Current Configuration on page 69](#)
- [Adding Configuration Statements and Identifiers on page 72](#)
- [Deleting a Statement from the Configuration on page 74](#)
- [Renaming an Identifier on page 76](#)
- [Inserting a New Identifier on page 77](#)
- [Copying a Configuration from One Configuration Location to Another on page 78](#)
- [Displaying Set Commands for the Configuration \(SRC CLI\) on page 79](#)

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.

- See Also**
- [Creating an SRC Configuration on page 65](#)
 - [Making Changes to the Configuration on page 17](#)
 - [Displaying the Current Configuration on page 69](#)
 - [Deleting a Statement from the Configuration on page 74](#)
 - [Reverting to a Previous SRC Configuration on page 128](#)

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:

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

- See Also**
- [Adding Configuration Statements and Identifiers on page 72](#)
 - [Deleting a Statement from the Configuration on page 74](#)
 - [Understanding SRC CLI Configuration Mode on page 61](#)
 - [Creating an SRC Configuration on page 65](#)

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;
      }
    }
  }
  . . .
}
```

- See Also**
- [Creating an SRC Configuration on page 65](#)
 - [Before You Load a Configuration on page 110](#)
 - [Commands to Load a Configuration on page 110](#)
 - [Committing a Configuration on page 80](#)
 - [Verifying a Configuration on page 80](#)

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 devices running Junos
OS
> 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>.
```

- See Also**
- [Understanding SRC CLI Configuration Mode on page 61](#)
 - [Cutting and Pasting Configuration Information at the SRC CLI on page 129](#)
 - [Renaming an Identifier on page 76](#)
 - [Inserting a New Identifier on page 77](#)
 - [Deleting a Statement from the Configuration on page 74](#)

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
```

- See Also**
- [Adding Configuration Statements and Identifiers on page 72](#)
 - [Understanding SRC CLI Configuration Mode on page 61](#)
 - [Creating an SRC Configuration on page 65](#)

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
```

- See Also**
- [Inserting a New Identifier on page 77](#)
 - [Adding Configuration Statements and Identifiers on page 72](#)
 - [Understanding SRC CLI Configuration Mode on page 61](#)
 - [Creating an SRC Configuration on page 65](#)

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 != "";
  }
}
```

- See Also**
- [Renaming an Identifier on page 76](#)
 - [Adding Configuration Statements and Identifiers on page 72](#)
 - [Understanding SRC CLI Configuration Mode on page 61](#)
 - [Creating an SRC Configuration on page 65](#)

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
```

- See Also**
- [Displaying the Current Configuration on page 69](#)
 - [Cutting and Pasting Configuration Information at the SRC CLI on page 129](#)
 - [Reverting to a Previous SRC Configuration on page 128](#)
 - [Committing a Configuration on page 80](#)
 - [How the SRC Configuration Is Stored on page 107](#)

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
```

- See Also**
- [How the SRC Configuration Is Stored on page 107](#)
 - [Understanding SRC CLI Configuration Mode on page 61](#)
 - [Creating an SRC Configuration on page 65](#)

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 17](#)
 - [Committing a Configuration on page 80](#)
 - [Displaying the Current Configuration on page 69](#)
 - [Verifying the Configuration for SRC Directory Access](#)
 - [Creating an SRC Configuration on page 65](#)

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 69](#)
- [Copying a Configuration from One Configuration Location to Another on page 78](#)
- [Exiting from Configuration Mode on page 66](#)
- [Verifying a Configuration on page 80](#)
- [Creating an SRC Configuration on page 65](#)

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 80](#)
- [Exiting from Configuration Mode on page 66](#)

- [Entering Configuration Mode on page 65](#)
- [Verifying a Configuration on page 80](#)
- [Creating an SRC Configuration on page 65](#)

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 68](#)
- [Entering Configuration Mode on page 65](#)
- [Viewing Information about Users Logged Into the SRC Software on page 59](#)
- [Creating an SRC Configuration on page 65](#)
- [How the SRC Configuration Is Stored on page 107](#)

CHAPTER 7

Filtering Command Output in the SRC CLI

- [Using Keyboard Sequences at the MORE Prompt in the SRC CLI on page 83](#)
- [Using the Pipe \(|\) Symbol When Entering Commands on page 85](#)
- [Using Regular Expressions with the Pipe Symbol on page 86](#)
- [Using Pipe Filter Functions on page 87](#)

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 14 on page 83](#) lists the keyboard sequences for the commands most frequently used at the **MORE** prompt.

Table 14: MORE Prompt Keyboard Sequences

Category	Action	Keyboard Sequence
Get Help	Display information about the keyboard sequences you can display at the MORE prompt. Commands listed that access files or the shell are not supported.	h, H

Table 14: MORE Prompt Keyboard Sequences (continued)

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.	F
	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 match filter when entering a command using the pipe symbol. See “Displaying Output That Matches a Regular Expression” on page 91.	
	Search backward for a string.	?string
	Repeat a previous search for a string.	Up arrow while in search mode
Interrupt or End Output, Redraw the Output, and Save the Output to a File	Repeat the previous search for a string in the opposite direction.	Down arrow while in search mode
	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 85](#)
 - [Saving Output to a File on page 92](#)

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 86](#)
 - [Displaying the End of the Output for a Command on page 90](#)
 - [Preventing Output from Being Paginated on page 92](#)
 - [Displaying Output in XML Tag Format on page 88](#)

- [Saving Output to a File on page 92](#)

Using Regular Expressions with the Pipe Symbol

The **except**, **find**, and **match** filters used with the pipe symbol employ regular expressions to filter output. Juniper Networks uses the regular expressions as defined in POSIX 1003.2. (See [Table 15 on page 86](#).) If the regular expression contains spaces, operators, or wildcard characters, enclose the expression in quotation marks.

Table 15: Common Regular Expression Operators in Operational Mode Commands

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

For example, if a command produces the following output:

```
12
22
3 2 1
4
```

A pipe filter of **| match 2** displays the following output:

```
12
22
3 2 1
```

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

```
22
4
```

Related Documentation

- [Using the Pipe \(|\) Symbol When Entering Commands on page 85](#)
- [Disregarding Output That Does Not Match a Regular Expression on page 89](#)

- [Displaying Output from the First Match of a Regular Expression on page 89](#)
- [Displaying Output That Matches a Regular Expression on page 91](#)
- [Saving Output to a File on page 92](#)

Using Pipe Filter Functions

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

- [Counting the Number of Lines of Output on page 87](#)
- [Displaying Summarized Output for a Hierarchy Level \(SRC CLI\) on page 87](#)
- [Displaying Output in XML Tag Format on page 88](#)
- [Disregarding Output That Does Not Match a Regular Expression on page 89](#)
- [Displaying Output from the First Match of a Regular Expression on page 89](#)
- [Displaying the End of the Output for a Command on page 90](#)
- [Displaying Output That Matches a Regular Expression on page 91](#)
- [Preventing Output from Being Paginated on page 92](#)
- [Saving Output to a File on page 92](#)

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
```

- See Also**
- [Displaying Output in XML Tag Format on page 88](#)
 - [Displaying the End of the Output for a Command on page 90](#)
 - [Preventing Output from Being Paginated on page 92](#)
 - [Saving Output to a File on page 92](#)
 - [Using the Pipe \(|\) Symbol When Entering Commands on page 85](#)

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;
```

- See Also**
- [Saving Output to a File on page 92](#)
 - [Displaying Output in XML Tag Format on page 88](#)
 - [Understanding SRC Command and Statement Hierarchies on page 4](#)
 - [Using the Pipe \(|\) Symbol When Entering Commands on page 85](#)

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>
```

- See Also**
- [Displaying Command Output on page 24](#)
 - [Counting the Number of Lines of Output on page 87](#)
 - [Displaying the End of the Output for a Command on page 90](#)
 - [Saving Output to a File on page 92](#)
 - [Using the Pipe \(|\) Symbol When Entering Commands on page 85](#)

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 *Common Regular Expression Operators in Operational Mode Commands* in “Using Regular Expressions with the Pipe Symbol” on page 86.

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;
    }
  }
}
```

- See Also**
- [Displaying Output from the First Match of a Regular Expression on page 89](#)
 - [Displaying Output That Matches a Regular Expression on page 91](#)
 - [Saving Output to a File on page 92](#)
 - [Using Regular Expressions with the Pipe Symbol on page 86](#)

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 *Common Regular Expression Operators in Operational Mode Commands* in “Using Regular Expressions with the Pipe Symbol” on page 86.

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 Core(s) 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%

. . .
```

- See Also**
- [Disregarding Output That Does Not Match a Regular Expression on page 89](#)
 - [Displaying Output That Matches a Regular Expression on page 91](#)
 - [Saving Output to a File on page 92](#)
 - [Using Regular Expressions with the Pipe Symbol on page 86](#)

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          30.000 degrees C
Physical CPU-1  34.000 degrees C
Physical CPU-2  29.000 degrees C
```

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

- See Also**
- [Counting the Number of Lines of Output on page 87](#)
 - [Displaying Output in XML Tag Format on page 88](#)
 - [Preventing Output from Being Paginated on page 92](#)
 - [Saving Output to a File on page 92](#)
 - [Using the Pipe \(|\) Symbol When Entering Commands on page 85](#)

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 *Common Regular Expression Operators in Operational Mode Commands* in “[Using Regular Expressions with the Pipe Symbol](#)” on page 86.

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 {
```

- See Also**
- [Disregarding Output That Does Not Match a Regular Expression on page 89](#)
 - [Displaying Output from the First Match of a Regular Expression on page 89](#)
 - [Saving Output to a File on page 92](#)

- [Using Regular Expressions with the Pipe Symbol on page 86](#)

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
```

- See Also**
- [Displaying Output in XML Tag Format on page 88](#)
 - [Counting the Number of Lines of Output on page 87](#)
 - [Displaying the End of the Output for a Command on page 90](#)
 - [Saving Output to a File on page 92](#)

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>
```

- See Also**
- [Counting the Number of Lines of Output on page 87](#)
 - [Displaying Output in XML Tag Format on page 88](#)
 - [Preventing Output from Being Paginated on page 92](#)
 - [Using the Pipe \(|\) Symbol When Entering Commands on page 85](#)

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 16 on page 95](#) lists some of the CLI keyboard sequences. They are the same as those used in Emacs.

Table 16: CLI Keyboard Sequences

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

Table 16: CLI Keyboard Sequences (continued)

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

PART 3

SRC CLI Environment and SRC Module and Component Management

- [Controlling the SRC CLI Environment on page 99](#)
- [Managing SRC Configurations on page 107](#)
- [Managing SRC Modules and Components with the CLI on page 131](#)

CHAPTER 9

Controlling the SRC CLI Environment

- [Commands to Control the SRC CLI Environment Overview on page 99](#)
- [Setting the Editing Level for the SRC CLI on page 100](#)
- [Setting the Terminal Type for the SRC CLI on page 101](#)
- [Setting the Language for the Terminal Environment for the SRC CLI on page 101](#)
- [Setting the Screen Length for the SRC CLI on page 102](#)
- [Setting the Screen Width for the SRC CLI on page 102](#)
- [Changing the Password for the SRC CLI on page 103](#)
- [Setting the SRC CLI Prompt on page 103](#)
- [Setting the Directory for the SRC CLI on page 104](#)
- [Setting Command Completion for the SRC CLI on page 104](#)
- [Viewing Settings for the SRC CLI on page 105](#)

Commands to Control the SRC CLI Environment Overview

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 100](#)
- [Setting Command Completion for the SRC CLI on page 104](#)
- [Changing the Password for the SRC CLI on page 103](#)
- [Viewing C Series Controller Information on page 54](#)
- [Commonly Used Operational Mode Commands on page 49](#)

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 17 on page 100](#) describes the editing levels.

Table 17: 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 25](#)
- [Commands to Control the SRC CLI Environment Overview on page 99](#)
- [Setting the Terminal Type for the SRC CLI on page 101](#)
- [Setting the SRC CLI Prompt on page 103](#)
- [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 101](#)
- [Setting the Screen Length for the SRC CLI on page 102](#)
- [Setting the Screen Width for the SRC CLI on page 102](#)
- [Viewing Settings for the SRC CLI on page 105](#)

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 101](#)
- [Setting the Screen Length for the SRC CLI on page 102](#)
- [Setting the Screen Width for the SRC CLI on page 102](#)
- [Setting the Editing Level for the SRC CLI on page 100](#)

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 102](#)
- [Setting the Language for the Terminal Environment for the SRC CLI on page 101](#)
- [Setting the Terminal Type for the SRC CLI on page 101](#)
- [Viewing Settings for the SRC CLI on page 105](#)
- [Setting the Editing Level for the SRC CLI on page 100](#)

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 102](#)
 - [Setting the Terminal Type for the SRC CLI on page 101](#)
 - [Setting the Language for the Terminal Environment for the SRC CLI on page 101](#)
 - [Viewing Settings for the SRC CLI on page 105](#)
 - [Setting the Editing Level for the SRC CLI on page 100](#)

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 101](#)
 - [Setting the SRC CLI Prompt on page 103](#)
 - [Commands to Control the SRC CLI Environment Overview on page 99](#)
 - [Setting the Editing Level for the SRC CLI on page 100](#)

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 28](#)

- [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 39](#)
- [Using Command Completion in Configuration Mode on page 40](#)

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 100](#)
- [Setting the Terminal Type for the SRC CLI on page 101](#)
- [Setting the Screen Length for the SRC CLI on page 102](#)
- [Setting the Screen Width for the SRC CLI on page 102](#)
- [Viewing Information About the SRC CLI \(C-Web Interface\)](#)

CHAPTER 10

Managing SRC Configurations

- [How the SRC Configuration Is Stored on page 107](#)
- [Commands to Save a Configuration to a File on page 108](#)
- [Updating the SRC Configuration on page 109](#)
- [About SRC Configuration Files in XML Format on page 113](#)
- [About SRC Configuration Files in Text Format on page 116](#)
- [About SRC Configuration Files in Set Format on page 118](#)
- [Preparing a File to Be Loaded into the Current SRC Configuration on page 119](#)
- [Loading an SRC Configuration on page 120](#)
- [Comparing SRC Configurations on page 127](#)
- [Reverting to a Previous SRC Configuration on page 128](#)
- [Cutting and Pasting Configuration Information at the SRC CLI on page 129](#)

How the SRC Configuration Is Stored

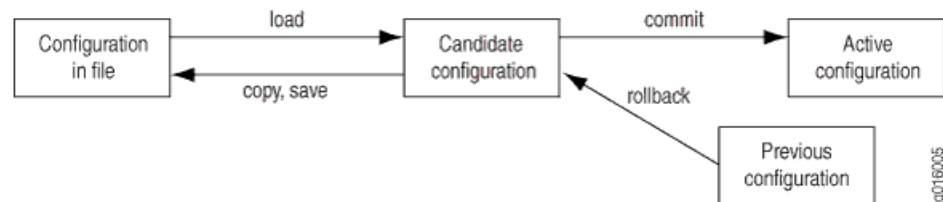
When you edit a configuration, you work in a copy of the current configuration to create a candidate configuration. Changes you make to the candidate configuration are visible in the CLI immediately. If multiple users edit the configuration at the same time, all users can see all changes.

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 108](#) 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 65](#)
- [SRC Configuration Updates on page 109](#)
- [About SRC Configuration Files in XML Format on page 113](#)
- [About SRC Configuration Files in Text Format on page 116](#)
- [About SRC Configuration Files in Set Format on page 118](#)
- [Cutting and Pasting Configuration Information at the SRC CLI on page 129](#)

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 107](#)

Updating the SRC Configuration

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

- [SRC Configuration Updates on page 109](#)
- [Before You Load a Configuration on page 110](#)
- [Commands to Load a Configuration on page 110](#)
- [Attributes in SRC Configuration Files on page 113](#)

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.

- See Also**
- [How the SRC Configuration Is Stored on page 107](#)
 - [Attributes in SRC Configuration Files on page 113](#)
 - [Replacing the Current Configuration with the Default SRC Configuration on page 120](#)
 - [Reverting to a Previous SRC Configuration on page 128](#)

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 configuration files, see:

- Text format—*SRC PE CLI Command Reference*
- XML format—*SRC XML API Configuration Reference*

You can save the configuration to text, XML, or set 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
```

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

```
[edit]
user@host# save backupcfg-set.txt format set
```

- See Also**
- [Cutting and Pasting Configuration Information at the SRC CLI on page 129](#)
 - [About SRC Configuration Files in XML Format on page 113](#)
 - [About SRC Configuration Files in Text Format on page 116](#)
 - [About SRC Configuration Files in Set Format on page 118](#)
 - [Commands to Load a Configuration on page 110](#)
 - [Preparing a File to Be Loaded into the Current SRC Configuration on page 119](#)

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.



NOTE: For a merge, replace, or override operation, you cannot use the file created by using the **save filename format set** command.

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.

- See Also**
- [Creating an SRC Configuration on page 65](#)
 - [Commands to Save a Configuration to a File on page 108](#)
 - [Before You Load a Configuration on page 110](#)
 - [Preparing a File to Be Loaded into the Current SRC Configuration on page 119](#)
 - [Loading a Configuration at a Specified Hierarchy Level on page 127](#)
 - [Cutting and Pasting Configuration Information at the SRC CLI on page 129](#)

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.

- See Also**
- [About SRC Configuration Files in XML Format on page 113](#)
 - [About SRC Configuration Files in Text Format on page 116](#)
 - [About SRC Configuration Files in Set Format on page 118](#)
 - [Preparing a File to Be Loaded into the Current SRC Configuration on page 119](#)

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>
<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 113](#)
- [About SRC Configuration Files in Text Format on page 116](#)
- [Preparing a File to Be Loaded into the Current SRC Configuration on page 119](#)

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 113](#)
 - [About SRC Configuration Files in XML Format on page 113](#)
 - [Preparing a File to Be Loaded into the Current SRC Configuration on page 119](#)

About SRC Configuration Files in Set Format

To create a configuration file in set format, you can perform the following tasks:

- Run the **save *filename* format set** command that saves the configurations to a file in set format.
- Run the **show** command at a specified hierarchy level, and then copy the output into a file.

You can also create a configuration file in set format 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 file.

The set format structure follows the same hierarchy as the CLI. For example, in configuration mode the following statements are available at the **[edit shared sae group *group-name* configuration]** hierarchy level:

```
[edit shared sae group group-name configuration]
user@host# set configuration driver aaa ?
Possible completions:
  keep-alive-timeout      Keepalive timeout (0..INF s)
  max-update-interval     Maximum interval of interim updates (0..INF s)
  registry-retry-interval Registry retry timeout (0..INF s)
  reply-timeout           Reply timeout (0..INF s)
  resume-unrecovered      Resume unrecovered user session
  sequential-message-timeout Sequential message timeout (0..INF s)
  thread-idle-timeout     Thread idle timeout (0..INF s)
  thread-pool-size        Thread pool size (100..400)
  transient-session-timeout Transient session timeout (0..INF s)
  update-grace-period      Grace period for an interim update (0..INF s)
```

In a set format configuration file, the configurations are the same as the statements in the **[edit shared sae group *group-name* configuration]** hierarchy.

```
set shared sae group POP-ID configuration driver aaa keep-alive-timeout 60
set shared sae group POP-ID configuration driver aaa max-update-interval 3600
set shared sae group POP-ID configuration driver aaa registry-retry-interval
30
set shared sae group POP-ID configuration driver aaa reply-timeout 25
set shared sae group POP-ID configuration driver aaa resume-unrecovered
set shared sae group POP-ID configuration driver aaa sequential-message-timeout
20
set shared sae group POP-ID configuration driver aaa thread-idle-timeout 60
set shared sae group POP-ID configuration driver aaa thread-pool-size 50
set shared sae group POP-ID configuration driver aaa transient-session-timeout
90
set shared sae group POP-ID configuration driver aaa update-grace-period 900
```

Related Documentation

- [Attributes in SRC Configuration Files on page 113](#)
- [About SRC Configuration Files in Text Format on page 116](#)
- [About SRC Configuration Files in XML Format on page 113](#)

- [Preparing a File to Be Loaded into the Current SRC Configuration on page 119](#)
- [Before You Load a Configuration on page 110](#)

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 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 and set 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 53](#).

- 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>
```

For information about enabling SSH on the C Series Controller, see *Enabling Remote Users to Access the C-Web Interface*.

**Related
Documentation**

- [Before You Load a Configuration on page 110](#)
- [About SRC Configuration Files in XML Format on page 113](#)
- [About SRC Configuration Files in Text Format on page 116](#)
- [About SRC Configuration Files in Set Format on page 118](#)
- [Commands to Load a Configuration on page 110](#)

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 120](#)
- [Merging the Active Configuration with Another Configuration on page 121](#)
- [Replacing the Configuration on page 123](#)
- [Replacing Parts of the Configuration on page 124](#)
- [Adding a Configuration Through Configuration Mode Commands on page 126](#)
- [Loading a Configuration at a Specified Hierarchy Level on page 127](#)

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.

- See Also**
- [Merging the Active Configuration with Another Configuration on page 121](#)
 - [Replacing the Configuration on page 123](#)
 - [Replacing Parts of the Configuration on page 124](#)
 - [Adding a Configuration Through Configuration Mode Commands on page 126](#)
 - [Reverting to a Previous SRC Configuration on page 128](#)

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 text or XML files. The examples in this section use the configurations in XML files.



NOTE: You cannot use the text file saved by using the **save filename** format set command to load the CLI configurations.

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 127](#).

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>
```

- See Also**
- [Replacing the Current Configuration with the Default SRC Configuration on page 120](#)
 - [Adding a Configuration Through Configuration Mode Commands on page 126](#)

- [Reverting to a Previous SRC Configuration on page 128](#)

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.



NOTE: You cannot use the file saved by using the `save filename format set` command to replace the CLI configurations.

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>
```

- See Also**
- [Replacing the Current Configuration with the Default SRC Configuration on page 120](#)
 - [Merging the Active Configuration with Another Configuration on page 121](#)
 - [Replacing Parts of the Configuration on page 124](#)
 - [Adding a Configuration Through Configuration Mode Commands on page 126](#)
 - [Reverting to a Previous SRC Configuration on page 128](#)

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



NOTE: You cannot replace the CLI configurations from a file saved in set 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 127.

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 113.

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 113.

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>
```

- See Also**
- [Replacing the Current Configuration with the Default SRC Configuration on page 120](#)
 - [Merging the Active Configuration with Another Configuration on page 121](#)
 - [Replacing the Configuration on page 123](#)
 - [Adding a Configuration Through Configuration Mode Commands on page 126](#)
 - [Reverting to a Previous SRC Configuration on page 128](#)

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
```

- See Also**
- [Replacing the Current Configuration with the Default SRC Configuration on page 120](#)
 - [Merging the Active Configuration with Another Configuration on page 121](#)
 - [Replacing the Configuration on page 123](#)
 - [Replacing Parts of the Configuration on page 124](#)
 - [Reverting to a Previous SRC Configuration on page 128](#)

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
```

- See Also**
- [Before You Load a Configuration on page 110](#)
 - [Adding a Configuration Through Configuration Mode Commands on page 126](#)
 - [Commands to Load a Configuration on page 110](#)

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}jW4yE2tHU5Euk";
+   }
+   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 65](#)
- [Types of SRC Commands and Statements on page 25](#)
- [How the SRC Configuration Is Stored on page 107](#)
- [Verifying a Configuration on page 80](#)

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 120](#)
- [Replacing the Configuration on page 123](#)
- [Replacing Parts of the Configuration on page 124](#)
- [Adding a Configuration Through Configuration Mode Commands on page 126](#)
- [Preparing a File to Be Loaded into the Current SRC Configuration on page 119](#)

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 121](#)
- [Replacing the Configuration on page 123](#)
- [Replacing Parts of the Configuration on page 124](#)
- [Adding a Configuration Through Configuration Mode Commands on page 126](#)
- [Loading a Configuration at a Specified Hierarchy Level on page 127](#)
- [How the SRC Configuration Is Stored on page 107](#)

- [Commands to Load a Configuration on page 110](#)

CHAPTER 11

Managing SRC Modules and Components with the CLI

- [Verifying Status of SRC Components on page 131](#)
- [Enabling SRC Components on page 132](#)
- [Disabling an SRC Component on page 133](#)
- [Restarting an SRC Component on page 134](#)

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
acp        Release: 7.8 Build: ACP.A.MAIN.1480      disabled
activity   Release: 7.8 Build: ACTIVITY.A.MAIN.1480   running
agent       Release: 7.8 Build: SYSMAN.A.MAIN.1480     disabled
appsvr     Release: 7.8 Build: JBOSS.A.MAIN.1480     disabled
cli         Release: MAIN Build: CLI.A.MAIN.1480       running
database   Release: 7.8 Build: SSR.A.MAIN.1480       disabled
diameter    Release: 7.8 Build: DIAMETER.A.MAIN.1480   running
dsa         Release: 7.8 Build: GATEWAYAPPS.A.MAIN.1480 disabled
editor      Release: 7.8 Build: EDITOR.A.MAIN.1480     running
extsubmon   Release: 7.8 Build: MONAGENT.A.MAIN.1480   disabled
gw-3gpp     Release: 7.8 Build: 3GPPGW.A.MAIN.1480    disabled
gy-3gpp     Release: 7.8 Build: 3GPPGY.A.MAIN.1480    running
ims         Release: 7.8 Build: IMS.A.MAIN.1480       disabled
jdb         Release: 7.8 Build: DIRXA.A.MAIN.1480     running
licSvr     Release: 7.8 Build: LICSVR.A.MAIN.1480    disabled
naming      Release: 7.8 Build: NAMING.A.MAIN.1480     running
nic         Release: 7.8 Build: GATEWAY.A.MAIN.1480    running
redir       Release: 7.8 Build: REDIR.A.MAIN.1480     disabled
sae         Release: 7.8 Build: SAE.A.MAIN.1480       running
sic         Release: 7.8 Build: SICCLI.A.MAIN.1480    disabled
vta         Release: 7.8 Build: VTA.A.MAIN.1480       disabled
webadm      Release: 7.8 Build: WEBADM.A.MAIN.1480    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
- Network Information Collector (NIC)
- Policy and Services Editor
- Redirect Server

- SNMP agent
- Volume Tracking Application (VTA)
- Third-Generation Partnership Project (3GPP) gateway
- 3GPP Gy

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 133](#)
- [Restarting an SRC Module or Component on page 56](#)
- [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
acp        Release: 7.8 Build: ACP.A.MAIN.1480      disabled
activity   Release: 7.8 Build: ACTIVITY.A.MAIN.1480  running
agent      Release: 7.8 Build: SYSMAN.A.MAIN.1480     disabled
appsvr     Release: 7.8 Build: JBOSS.A.MAIN.1480     disabled
cli        Release: MAIN Build: CLI.A.MAIN.1480       running
database   Release: 7.8 Build: SSR.A.MAIN.1480       disabled
diameter   Release: 7.8 Build: DIAMETER.A.MAIN.1480  running
dsa        Release: 7.8 Build: GATEWAYAPPS.A.MAIN.1480 disabled
editor     Release: 7.8 Build: EDITOR.A.MAIN.1480    running
extsubmon  Release: 7.8 Build: MONAGENT.A.MAIN.1480  disabled
gw-3gpp    Release: 7.8 Build: 3GPPGW.A.MAIN.1480    disabled
gy-3gpp    Release: 7.8 Build: 3GPPGY.A.MAIN.1480    running
ims        Release: 7.8 Build: IMS.A.MAIN.1480       disabled
jdb        Release: 7.8 Build: DIRXA.A.MAIN.1480     running
licSvr     Release: 7.8 Build: LICSVR.A.MAIN.1480    disabled
naming     Release: 7.8 Build: NAMING.A.MAIN.1480    running
nic        Release: 7.8 Build: GATEWAY.A.MAIN.1480    running
redir      Release: 7.8 Build: REDIR.A.MAIN.1480     disabled
sae        Release: 7.8 Build: SAE.A.MAIN.1480       running
sic        Release: 7.8 Build: SICCLI.A.MAIN.1480     disabled
vta        Release: 7.8 Build: VTA.A.MAIN.1480       disabled
webadm     Release: 7.8 Build: WEBADM.A.MAIN.1480    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 132](#)
- [Restarting an SRC Module or Component on page 56](#)
- [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 132](#)
- [Disabling an SRC Component on page 133](#)
- [Configuring SRC Components](#)