



SRC PE Software

Network Guide

Release

4.0.x



Published: 2010-07-29

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

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

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

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

SRC PE Software Network Guide
Release 4.0.x
Copyright © 2010, Juniper Networks, Inc.
All rights reserved. Printed in USA.

Writing: Linda Creed, Justine Kangas, Betty Lew, Helen Shaw
Editing: Fran Mues
Illustration: Nathaniel Woodward
Cover Design: Edmonds Design

Revision History
July 2010—Revision 2

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

YEAR 2000 NOTICE

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

SOFTWARE LICENSE

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

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

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

END USER LICENSE AGREEMENT

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

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

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

- a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.
- b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.
- c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.
- d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.
- e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

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

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

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

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

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

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

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

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

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

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

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

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

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

Abbreviated Table of Contents

	About the Documentation	xxix
Part 1	Operating the SAE	
Chapter 1	Overview of the SAE	3
Chapter 2	Configuring the SAE (SRC CLI)	13
Chapter 3	Managing Subscriber and Service Session Data (SRC CLI)	23
Chapter 4	Managing SAE Data (SRC CLI)	31
Chapter 5	Managing SAE Data (C-Web Interface)	37
Part 2	Using Juniper Networks Routers in the SRC Network	
Chapter 6	Using JUNOS Routers in the SRC Network (SRC CLI)	45
Chapter 7	Using JUNOS Routing Platforms in the SRC Network (SRC CLI)	65
Part 3	Using Network Devices in the SRC Network	
Chapter 8	Integrating Third-Party Network Devices into the SRC Network (SRC CLI)	91
Part 4	Locating Subscriber Management Information	
Chapter 9	Locating Subscriber Information with the NIC	111
Chapter 10	Configuring the NIC (SRC CLI)	127
Chapter 11	Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrflp	149
Chapter 12	Configuring Applications to Communicate with an SAE	163
Chapter 13	Configuring SRC Applications to Communicate with an SAE (SRC CLI)	165
Chapter 14	Developing Applications That Use NIC	173
Chapter 15	NIC Resolution Process	181
Chapter 16	NIC Configuration Scenarios	187
Part 5	Providing Admission Control with SRC ACP	
Chapter 17	Overview of Providing Admission Control with SRC ACP	223
Chapter 18	Configuring Admission Control (SRC CLI)	233
Chapter 19	Configuring Congestion Point Classification (SRC CLI)	269
Chapter 20	Managing SRC ACP (SRC CLI)	279

Chapter 21	Monitoring Admission Control (SRC CLI)	283
Chapter 22	Monitoring Admission Control (C-Web Interface)	293
Part 6	Using External Subscriber Monitor	
Chapter 23	Configuring External Subscriber Monitor with the SRC CLI	313
Chapter 24	Monitoring External Subscriber Events with the SRC CLI	325
Chapter 25	Monitoring External Subscriber Events with the C-Web Interface	329
Part 7	Using Session State Registrar	
Chapter 26	Session State Registrar Overview	333
Chapter 27	Planning Your Session State Registrar Cluster	355
Chapter 28	Configuring the Session State Registrar (SRC CLI)	363
Chapter 29	Managing the SSR Cluster	381
Chapter 30	Monitoring the SSR Cluster	385
Part 8	Using the Subscriber Information Collector	
Chapter 31	Overview of the Subscriber Information Collector	393
Chapter 32	Configuring the Subscriber Information Collector with the SRC CLI	415
Chapter 33	Monitoring the Subscriber Information Collector with the SRC CLI	461
Part 9	Index	
	Index	465

Table of Contents

	About the Documentation	xxix
	SRC Documentation and Release Notes	xxix
	Audience	xxix
	Documentation Conventions	xxix
	Documentation Feedback	xxx
	Requesting Technical Support	xxx
	Self-Help Online Tools and Resources	xxxii
	Opening a Case with JTAC	xxxii
Part 1	Operating the SAE	
Chapter 1	Overview of the SAE	3
	Role of the SAE	3
	Connections to Managed Devices	3
	COPS Connection Between JUNOS Routers and the SAE	4
	Beep Connection Between JUNOS Routing Platforms and the SAE	4
	COPS Connection Between CMTS Devices and the SAE	4
	COPS Connection Between Juniper Policy Servers and the SAE	5
	SAE Plug-Ins	5
	Internal Plug-Ins	6
	External Plug-Ins	6
	Hosted Plug-Ins	7
	Tracking and Controlling Subscriber and Service Sessions with SAE APIs	7
	SAE Core API	8
	SAE CORBA Remote API	8
	SAE Accounting	9
	Accounting Policy	10
	Subscription Process	11
	Tracking Subscriber Sessions	11
	Accounting Plug-Ins	11
	Interim Accounting	11
Chapter 2	Configuring the SAE (SRC CLI)	13
	SRC Access to Directory Data	13
	Configuring LDAP Access to Directory Data (SRC CLI)	14
	Configuring Access Through LDAPS to Service and Subscriber Data (SRC CLI)	14
	Configuring Access to Subscriber Data (SRC CLI)	15
	Configuring Access to Service Data (SRC CLI)	17
	Configuring Access to Policy Data (SRC CLI)	18
	Configuring Access to the Persistent Login Cache (SRC CLI)	19

	Configuring the Location of Network Device Data (SRC CLI)	20
	Enabling Automatic Discovery of Changes in SAE Configuration Data (SRC CLI)	21
	Setting the Timeout and Number of Events for SAE Directory Eventing (SRC CLI)	22
Chapter 3	Managing Subscriber and Service Session Data (SRC CLI)	23
	Storing Subscriber and Service Session Data	23
	Session Store Files	23
	Active and Passive Session Stores	23
	Standby SAEs	24
	Session Store File Rotation	24
	Configuring the Session Store Feature (SRC CLI)	25
	Configuring Session Store Parameters for a Device Driver	25
	Configuring Global Session Store Parameters	27
	Reducing the Size of Objects for the Session Store Feature	28
	Configuring the Number of Threads for Sessions (SRC CLI)	29
Chapter 4	Managing SAE Data (SRC CLI)	31
	Commands to Manage SAE Data	31
	Reloading the SAE Data (SRC CLI)	32
	Reloading the SAE Configuration (SRC CLI)	32
	Reloading Services (SRC CLI)	33
	Reloading Subscriptions (SRC CLI)	33
	Reloading Interface Classification Scripts (SRC CLI)	33
	Reloading Domain Maps (SRC CLI)	33
	Removing the Directory Blacklist (SRC CLI)	34
	Removing Login Registrations (SRC CLI)	34
	Removing Equipment Registrations (SRC CLI)	35
	Modifying Failover Server Parameters (SRC CLI)	35
	Shutting Down the Device Drivers (SRC CLI)	36
Chapter 5	Managing SAE Data (C-Web Interface)	37
	Reloading the SAE Data (C-Web Interface)	37
	Reloading the SAE Configuration (C-Web Interface)	37
	Reloading Services (C-Web Interface)	38
	Reloading Subscriptions (C-Web Interface)	38
	Reloading Interface Classification Scripts (C-Web Interface)	39
	Reloading Domain Maps (C-Web Interface)	39
	Removing the Directory Blacklist (C-Web Interface)	39
	Removing Login Registrations (C-Web Interface)	40
	Removing Equipment Registrations (C-Web Interface)	40
	Modifying Failover Server Parameters (C-Web Interface)	41
	Shutting Down the Device Drivers (C-Web Interface)	41

Part 2

Using Juniper Networks Routers in the SRC Network

Chapter 6

Using JUNOS Routers in the SRC Network (SRC CLI) 45

COPS Connection Between JUNOS Routers and the SAE	45
Highly Available Connections to JUNOS Routers	46
Adding JUNOS Routers and Virtual Routers (SRC CLI)	46
Adding Operative JUNOS Routers and Virtual Routers	47
Adding Routers Individually (SRC CLI)	47
Adding Virtual Routers Individually (SRC CLI)	48
Configuring the SAE to Manage JUNOS Routers (SRC CLI)	50
How SNMP Obtains Information from Routers for the SRC Software	52
Developing Router Initialization Scripts for Network Devices and Juniper Networks	
Routers	53
Interface Object Fields	53
Required Methods	55
Example: Router Initialization Script	55
Specifying JUNOS Router Initialization Scripts on the SAE (SRC CLI)	55
Accessing the Router CLI	57
Starting the SRC Client on a JUNOS Router	57
Stopping the SRC Client on a JUNOS Router	58
Monitoring Interactions Between the SAE and the Router Running JUNOS	
Software	58
Troubleshooting Problems with Managing JUNOS Routers	59
Viewing the State of JUNOS Device Drivers (SRC CLI)	60
Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI)	61
Viewing Statistics for All JUNOS Device Drivers (SRC CLI)	61
Viewing the State of JUNOS Device Drivers (C-Web Interface)	62
Viewing Statistics for All JUNOS Device Drivers (C-Web Interface)	62

Chapter 7

Using JUNOS Routing Platforms in the SRC Network (SRC CLI) 65

BEEP Connection Between JUNOS Routing Platforms and the SAE	65
Adding JUNOS Routing Platforms and Virtual Routers (SRC CLI)	66
Adding Operative JUNOS Routing Platforms (SRC CLI)	66
Adding Routers Individually (SRC CLI)	67
Adding Virtual Routers Individually (SRC CLI)	68
Configuring the SAE to Manage JUNOS Routing Platforms (SRC CLI)	69
Configuring Secure Connections Between the SAE and JUNOS Routing	
Platforms	72
Adding the Server Certificate on the Routing Platform	72
Creating a Client Certificate for the Router	73
Adding the Client Certificate on the Router	73
Configuring the SAE to Use TLS (SRC CLI)	74
Configuring TLS on the SAE (SRC CLI)	74
SAE Verification of JUNOS Configuration Changes	75
Setting Up Periodic Configuration Checking (SRC CLI)	76
Using SNMP to Retrieve Information from JUNOS Routers and JUNOS Routing	
Platforms (SRC CLI)	76
Specifying Router Initialization Scripts on the SAE (SRC CLI)	77

Configuring JUNOS Routing Platforms to Interact with the SAE	78
SAE Tracking for LSPs Configured on JUNOS Routing Platforms	79
Overview of SAE Tracking for LSPs Configured on JUNOS Routing Platforms	79
Configuring Event Tracking for JUNOS LSPs (SRC CLI)	80
Configuring the JUNOS Routing Platform to Apply Changes It Receives from the SAE	80
Disabling Interactions Between the SAE and JUNOS Routing Platforms	81
Monitoring Interactions Between the SAE and JUNOS Routing Platforms	81
Troubleshooting Problems Between the SRC module and JUNOS Device Drivers	82
Troubleshooting Problems with the SRC Software Process	82
Viewing the State of JUNOS Device Drivers (SRC CLI)	83
Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI)	83
Viewing Statistics for All JUNOS Device Drivers (SRC CLI)	84
Viewing the State of JUNOS Device Drivers (C-Web Interface)	85
Viewing Statistics for Specific JUNOS Device Drivers (C-Web Interface) ...	86
Viewing Statistics for All JUNOS Device Drivers (C-Web Interface)	86

Part 3

Chapter 8

Using Network Devices in the SRC Network

Integrating Third-Party Network Devices into the SRC Network (SRC CLI)	91
Overview of Integrating Network Devices into the SRC Network	91
SAE Communities	92
Storing Session Data	92
Using Script Services to Provision Third-Party Devices	92
Logging In Subscribers and Creating Sessions	93
Assigned IP Subscribers	93
Login Interactions with Assigned IP Subscribers	94
Event Notification from an IP Address Manager	95
Login with Event Notification	95
Configuration Tasks for Integrating Third-Party Network Devices (SRC CLI) ...	96
Setting Up Script Services	97
Adding Objects for Network Devices (SRC CLI)	98
Adding Virtual Router Objects (SRC CLI)	99
Setting Up SAE Communities (SRC CLI)	100
Configuring the SAE Community Manager	100
Specifying the Community Manager in the SAE Device Driver	101
Configuring SAE Properties for the Event Notification API (SRC CLI)	102
Developing Router Initialization Scripts for Network Devices and Juniper Networks	
Routers	103
Interface Object Fields	104
Required Methods	105
Example: Router Initialization Script	105
Copying Initialization Scripts to the C Series Controller	106
Specifying Initialization Scripts on the SAE (SRC CLI)	106
Using SNMP to Retrieve Information from Network Devices	107
Configuring Global SNMP Communities in the SRC Software (SRC CLI)	107

	Using the NIC Resolver in Environments That Have Third-Party Devices (SRC CLI)	108
Part 4	Locating Subscriber Management Information	
Chapter 9	Locating Subscriber Information with the NIC	111
	Locating Subscriber Management Information	111
	NIC Client/Server Mode	112
	NIC Local Host Mode	112
	Mapping Subscribers to a Managing SAE	113
	NIC Proxies and NIC Locators	113
	NIC Hosts	113
	NIC Agents	114
	NIC Resolvers	114
	High Availability for NIC	114
	High Availability in Existing NIC Configurations	115
	NIC Replication	115
	Planning a NIC Implementation	117
	NIC Configuration Scenarios	118
	NIC Agents Used in the NIC Configuration Scenarios	122
	Router Initialization Scripts with NIC Configuration Scenarios	124
Chapter 10	Configuring the NIC (SRC CLI)	127
	Configuration Statements for the NIC	127
	Configuration Statements for NIC Operating Properties	128
	Configuration Statements for NIC Scenarios	128
	Configuration Statements for NIC Logging	129
	Before You Configure the NIC	129
	Configuring the NIC (SRC CLI)	130
	Reviewing and Changing Operating Properties for the NIC (SRC CLI)	131
	Reviewing the Default NIC Operating Properties	131
	Changing NIC Operating Properties	132
	Configuring NIC Replication (SRC CLI)	133
	Configuring a NIC Scenario (SRC CLI)	134
	Defining the NIC Configuration to Use	134
	Configuring Directory Agents	137
	Configuring SAE Client Agents	139
	Configuring SAE Plug-In Agents	140
	Configuring the SAE to Communicate with SAE Plug-In Agents When You Use NIC Replication	141
	Configuring Advanced NIC Features	143
	Verifying Configuration for the NIC (SRC CLI)	143
	Starting the NIC (SRC CLI)	144
	Testing a NIC Resolution (SRC CLI)	144
	Stopping a NIC Host on a C Series Controller (SRC CLI)	145
	Restarting the NIC (SRC CLI)	145
	Restarting a NIC Agent (SRC CLI)	146
	Restarting a NIC Resolver (SRC CLI)	146
	Changing NIC Configurations (SRC CLI)	147

Chapter 11	Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrflp	149
	Overview of the Network Publisher	149
	NIC Document That Maps Subscriber IP Addresses to a JUNOS Interface	150
	Configuration Statements for the Network Publisher	150
	Before You Configure and Run the Network Publisher	151
	Configuring the Network Publisher (SRC CLI)	152
	Configuring Local Configuration for the Network Publisher	152
	Configuring Connections Between JUNOS Routing Platforms and the Network Publisher	153
	Configuring Router Authentication for the Network Publisher	154
	Configuring Routing Table Filters for the Network Publisher	155
	Configuring the Connection Between the Network Publisher and the Juniper Networks Database	156
	Running the Network Publisher (SRC CLI)	157
	Files Used to Test Network Publisher	158
	Configuring Information to Test the Network Publisher (SRC CLI)	158
	Troubleshooting Network Publisher Operations (SRC CLI)	159
	Reviewing the Information Collected from a JUNOS Routing Platform (SRC CLI)	160
Chapter 12	Configuring Applications to Communicate with an SAE	163
	Overview of NIC Proxy Configuration	163
	Before You Configure a NIC Proxy	164
Chapter 13	Configuring SRC Applications to Communicate with an SAE (SRC CLI)	165
	Configuration Statements for NIC Proxies	165
	Configuring Resolution Information for a NIC Proxy (SRC CLI)	166
	Changing the Configuration for the NIC Proxy Cache (SRC CLI)	168
	Configuring a NIC Proxy for NIC Replication (SRC CLI)	169
	Configuring NIC Test Data (SRC CLI)	171
Chapter 14	Developing Applications That Use NIC	173
	External Application Requirements for NIC	173
	External Non-Java Applications That Use NIC	173
	Creating a NIC Locator to Include with a Non-Java Application	174
	External Java Applications That Use NIC	175
	Developing a Java Application to Communicate with a NIC Proxy	176
	Instantiating a Configuration Manager	176
	Passing a Reference to the Configuration Manager to the NIC Factory	176
	Instantiating the NIC Factory Class	176
	Initializing Logging	177
	Instantiating the NIC Proxy	177
	Managing a Resolution Request	178
	Deleting Invalid Results from the NIC Proxy's Cache	179
	Removing the NIC Proxies	180
	Updating Information About Address Pools	180

Chapter 15	NIC Resolution Process	181
	Overview of the NIC Resolution Process	181
	NIC Realms	181
	Key to Value Resolution	182
	NIC Data Types	182
	Constraints as NIC Data Types	184
Chapter 16	NIC Configuration Scenarios	187
	Overview of NIC Configuration Scenarios	187
	OnePop Scenario	188
	Centralized Configuration	188
	Distributed Configuration	189
	Redundancy	189
	OnePopPcmm Scenario	190
	Centralized Configuration	191
	Distributed Configuration	192
	OnePopDynamicIp Scenario	192
	Centralized Configuration	193
	Distributed Configuration	194
	OnePopSharedIp Scenario	194
	Centralized Configuration	195
	Distributed Configuration	196
	OnePopStaticRouteIp	196
	Centralized Configuration	197
	Distributed Configuration	198
	OnePopVrflp Scenario	199
	Centralized Configuration	199
	Distributed Configuration	200
	OnePopAcctId Scenario	201
	OnePopLogin Scenario	203
	Centralized Configuration	204
	Distributed Configuration	205
	OnePopLoginPull Scenario	205
	OnePopPrimaryUser	206
	Centralized Configuration	207
	Distributed Configuration	207
	OnePopDnSharedIp Scenario	208
	Centralized Configuration	209
	Distributed Configuration	209
	OnePopAllRealms Scenario	212
	MultiPop Scenario	216
	IP Realm	217
	Shared IP Realm	218
	DN Realm	219

Part 5**Providing Admission Control with SRC ACP****Chapter 17****Overview of Providing Admission Control with SRC ACP 223**

Overview of SRC ACP	223
Deriving Congestion Points Automatically	225
Deriving Edge Congestion Points	225
Deriving Congestion Points from a Profile	226
Deriving Backbone Congestion Points	226
Allocating Bandwidth to Applications Not Controlled by SRC ACP	227
Use of Multiple SRC ACPs	228
Interactions Between SRC ACP and Other Components	228
Redundancy and State Synchronization	230
Fault Recovery	231
Creating an Application to Update Information for SRC ACP	231

Chapter 18**Configuring Admission Control (SRC CLI) 233**

Configuration Statements for SRC ACP	233
Configuring SRC ACP (SRC CLI)	236
Creating Grouped Configurations for SRC ACP (SRC CLI)	236
Configuring Local Properties for SRC ACP (SRC CLI)	237
Configuring Basic Local Properties for SRC ACP	237
Configuring Initial Properties for SRC ACP	238
Configuring Directory Connection Properties for SRC ACP	239
Configuring Initial Directory Eventing Properties for SRC ACP	240
Configuring the SAE for SRC ACP (SRC CLI)	241
Configuring SRC ACP as an External Plug-In	241
Configuring Event Publishers	241
Configuring the SAE to Monitor Interfaces for Congestion Points	241
Configuring SRC ACP Properties (SRC CLI)	243
Configuring Logging Destinations for SRC ACP	243
Configuring SRC ACP Operation	244
Configuring CORBA Interfaces	248
Configuring SRC ACP Redundancy	249
Configuring Connections to the Subscribers' Directory	250
Configuring Connections to the Services' Directory	252
Configuring SRC ACP Scripts and Classification	254
Configuring SRC ACP to Manage the Edge Network (SRC CLI)	255
Configuring Network Interfaces in the Directory for the Edge Network	255
Configuring Bandwidths for Subscribers	256
Assigning Network Interfaces to Subscribers	257
Configuring Bandwidths for Services in the Edge Network	258
Configuring SRC ACP to Manage the Backbone Network (SRC CLI)	259
Configuring Network Interfaces in the Directory for the Backbone Network	259
Extending SRC ACP Congestion Points for the Backbone Network	259
Configuring Action Congestion Points	260
Configuring Bandwidths for Services in the Backbone Network	261
Configuring Congestion Points for Services in the Backbone Network	261
Plug-In Attributes for Use with Backbone Congestion Point Expressions	263

	Using Functions for Backbone Congestion Point Classification Scripts	266
	Configuring Congestion Point Profiles in the Directory	267
	Assigning Interfaces to Congestion Point Profiles	267
Chapter 19	Configuring Congestion Point Classification (SRC CLI)	269
	Overview of Congestion Point Classification	269
	Congestion Point Classification Scripts	269
	Congestion Point Profiles	270
	Configuration Statements for Congestion Point Classification	270
	Classifying Congestion Points (SRC CLI)	270
	Configuring Targets and Criteria for Classification Scripts	270
	Configuring Classification Scripts Contents for Classification Scripts	271
	Configuring Congestion Point Classification Targets	271
	Congestion Point Classification Criteria	272
	Defining a Congestion Point Profile (SRC CLI)	276
	Congestion Point Expressions	276
	Expressions in Templates for Congestion Point Profiles	277
	Methods for Use with Scripting Expressions	277
	Match Criteria for Congestion Point Classification	278
Chapter 20	Managing SRC ACP (SRC CLI)	279
	Starting SRC ACP	279
	Stopping SRC ACP	279
	Reorganizing the File That Contains ACP Data	280
	Modifying Congestion Points	280
Chapter 21	Monitoring Admission Control (SRC CLI)	283
	Viewing Information About Subscriber Sessions in the Edge Network (SRC CLI)	283
	Viewing Edge Congestion Point Information by DN (SRC CLI)	284
	Viewing Edge Congestion Point Information by Subscriber Session (SRC CLI)	285
	Viewing Information About Services in the Backbone Network (SRC CLI)	285
	Viewing Backbone Congestion Point Information by DN (SRC CLI)	286
	Viewing Backbone Congestion Point Information by Service (SRC CLI)	287
	Viewing Action Congestion Point Information by Service (SRC CLI)	287
	Viewing Action Congestion Point Information by Congestion Point (SRC CLI)	288
	Viewing Information About Subscribers Obtained from External Applications (SRC CLI)	289
	Viewing Congestion Point Information by DN (SRC CLI)	290
	Viewing Congestion Point Information by Name (SRC CLI)	290
	Viewing SNMP Information for Devices (SRC CLI)	291
	Viewing SNMP Information for the Directory (SRC CLI)	291
	Viewing SNMP Information for SRC ACP (SRC CLI)	292
Chapter 22	Monitoring Admission Control (C-Web Interface)	293
	Viewing Information About Subscriber Sessions in the Edge Network (C-Web Interface)	293
	Viewing Information About Edge Congestion Points by DN (C-Web Interface)	294

Viewing Information About Edge Congestion Points by Subscriber Session (C-Web Interface)	295
Viewing Information About Services in a Backbone Network (C-Web Interface)	296
Viewing Information About Congestion Points in a Backbone Network by Expression (C-Web Interface)	298
Viewing Information About Congestion Points in a Backbone Network by DN (C-Web Interface)	299
Viewing Information about Action Congestion Points in a Backbone Network by Service (C-Web Interface)	300
Viewing Information about Action Congestion Points in a Backbone Network by Expression (C-Web Interface)	302
Viewing Information About Subscribers Obtained from External Applications (C-Web Interface)	304
Viewing Information About Congestion Points from an External Application by DN (C-Web Interface)	305
Viewing Information About Congestion Points from an External Application by Interface Name (C-Web Interface)	306
Viewing Statistics for the SRC ACP Configuration (C-Web Interface)	307
Viewing General Statistics for SRC ACP (C-Web Interface)	308
Viewing Statistics for the SRC ACP Directory (C-Web Interface)	308
Viewing Device Statistics for SRC ACP (C-Web Interface)	309

Part 6

Using External Subscriber Monitor

Chapter 23

Configuring External Subscriber Monitor with the SRC CLI	313
Overview of External Subscriber Monitor	313
Configuring External Subscriber Monitor (SRC CLI)	314
Configuring Basic Local Properties for External Subscriber Monitor	314
Configuring Initial Properties for External Subscriber Monitor	315
Configuring Directory Connection Properties for External Subscriber Monitor	315
Configuring Eventing Properties for External Subscriber Monitor	316
Configuring Logging Destinations for External Subscriber Monitor	316
Configuring the NIC Proxy for the Pseudo-RADIUS Server (SRC CLI)	318
Configuring Resolution Information for a NIC Proxy	318
Changing the Configuration for the NIC Proxy Cache	318
Configuring a NIC Proxy for NIC Replication	319
Configuring the Pseudo-RADIUS Server for External Subscriber Monitor (SRC CLI)	320
Configuring the Client Secret for External Subscriber Monitor (SRC CLI)	321
Configuring Event Notification for External Subscriber Monitor (SRC CLI)	322
Starting External Subscriber Monitor (SRC CLI)	323
Stopping External Subscriber Monitor (SRC CLI)	323

Chapter 24

Monitoring External Subscriber Events with the SRC CLI	325
Viewing Statistics for External Subscriber Monitor (SRC CLI)	325
Monitoring Statistics for External Subscriber Monitor (SRC CLI)	326
Viewing Statistics for External Subscriber Monitor Event Notifications (SRC CLI)	326

	Monitoring Statistics for External Subscriber Monitor Event Notifications (SRC CLI)	327
	Viewing Statistics for the Agent Process (SRC CLI)	328
Chapter 25	Monitoring External Subscriber Events with the C-Web Interface	329
	Viewing Statistics for External Subscriber Monitor (C-Web Interface)	329
	Viewing Statistics for External Subscriber Monitor Event Notifications (C-Web Interface)	330
	Viewing Statistics for the Agent Process (C-Web Interface)	330
Part 7	Using Session State Registrar	
Chapter 26	Session State Registrar Overview	333
	Overview of the Session State Registrar	333
	SSR Node Types	334
	SSR Node Groups	335
	C Series Controller Requirements	336
	SSR Cluster Configurations Overview	337
	Scaling the SSR Cluster	338
	Scaling the Front End of the Cluster	338
	Scaling the Back End of the Cluster	338
	SSR Cluster Network Requirements	339
	Supported SSR Cluster Configurations	341
	Failover Overview	342
	Failover Examples	342
	Possible Failure Scenarios	343
	Distributed Cluster Failure and Recovery	344
	SSR Database Schema	347
	Subscriber Sessions Table	347
	Attribute Associations	349
	Service Sessions Table	349
	Overview of Making Modifications to the SSR Database Schema	351
	SSR Database Operating Modes	351
	Distributing the SSR Cluster Configuration and Enabling SSR Client Components	352
	Enabling, Restarting and Disabling the SSR Component Database	352
Chapter 27	Planning Your Session State Registrar Cluster	355
	Planning the SSR Cluster Topology	355
	Identifying the C Series Controllers in the SSR Cluster	355
	SSR Cluster Planning Worksheets	357
Chapter 28	Configuring the Session State Registrar (SRC CLI)	363
	Configuration Changes and Their Impact on the SSR Cluster	363
	Configuration Statements for the SSR Cluster	364
	Configuring the Initial SSR Cluster (SRC CLI)	365
	Configuring the SSR Cluster ID (SRC CLI)	366
	Configuring the SSR Cluster Geometry (SRC CLI)	367
	Configuring the Nodes in the SSR Cluster (SRC CLI)	368
	Configuring the Management Servers in the SSR Cluster (SRC CLI)	369

	Configuring the Fields in the Subscriber Sessions Table (SRC CLI)	370
	Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table (SRC CLI)	371
	Modifying the SSR Database Schema in an Active Cluster (SRC CLI)	372
	Modifying Attribute Mapping in an Active SSR Cluster (SRC CLI)	373
	Adding Data Nodes to an Active SSR Cluster (SRC CLI)	374
	Adding Client Nodes to an Active SSR Cluster (SRC CLI)	375
	Adding a Management Server to an Active SSR Cluster	377
	Removing Data Nodes from an Active SSR Cluster	378
	Removing a Client Node from an Active SSR Cluster	379
	Removing a Management Server from an Active SSR Cluster	380
Chapter 29	Managing the SSR Cluster	381
	Placing the SSR Database into Maintenance Mode (SRC CLI)	381
	Placing the SSR Database into Production Mode (SRC CLI)	381
	Deleting the SSR Database (SRC CLI)	382
	Creating the SSR Database (SRC CLI)	382
	Enabling the SSR Database (SRC CLI)	383
	Disabling the SSR Database (SRC CLI)	383
	Restarting the SSR Database (SRC CLI)	383
	Deleting All Subscriber Sessions in the SSR Database	384
	Deleting Subscriber Sessions in the SSR Database By IP Address	384
Chapter 30	Monitoring the SSR Cluster	385
	Viewing the SSR Database Mode (SRC CLI)	385
	Viewing the Status of the SSR Cluster (SRC CLI)	385
	Viewing the Running Configuration of the SSR Database (SRC CLI)	386
	Viewing All Subscriber Sessions in the SSR Database (SRC CLI)	387
	Viewing Subscriber Sessions in the SSR Database by IP Address (SRC CLI)	388
	Viewing Subscriber Sessions in the SSR Database by Indexed Field (SRC CLI)	389
	Viewing the Total Number of Subscriber Sessions in the SSR Database (SRC CLI)	390
Part 8	Using the Subscriber Information Collector	
Chapter 31	Overview of the Subscriber Information Collector	393
	Subscriber Information Collector Overview	393
	Overview of Local and Shared Configurations for the SIC (SRC CLI)	394
	Local Configuration	394
	Shared Configuration	395
	Overview of SIC Accounting Methods and Targets (SRC CLI)	396
	Using the SSR Database as the Accounting Method	396
	Mapping Attributes When Using the Database Accounting Method	397
	Using the Proxy RADIUS Accounting Method	398
	Overview of SIC Request Routing Rules (SRC CLI)	399
	Explicit Routing Rules	399
	Implicit Routing Rules	401
	SIC Editing Rules Overview (SRC CLI)	401

	Overview of RADIUS Configuration for the SIC (SRC CLI)	404
	Accounting Listener	404
	Outbound Transport for the SIC Group	405
	Inbound and Outbound Transport for the SIC Server	405
	Network Elements	405
	Overview of Configuring Upstream Network Elements	405
	Overview of Configuring Downstream Network Elements	406
	Using the Proxy Function to Define Implicit Routing Rules	406
	Overview of RADIUS Transports for the SIC Group and Server	407
	Overview of SIC Dictionaries and Device Models (SRC CLI)	407
	Dictionaries and the Device Models Supported by the SIC Group	407
	Overview of Configuring Device Models and Their Associated Dictionaries for the SIC Group	408
	Modifying a Dictionary	408
	Overview of Configuring the Dictionaries Used By the SIC Group	408
	Overview of Loading SIC Dictionaries (SRC CLI)	408
	Overview of SIC Local Realms	409
	Overview of SIC Event Logging (SRC CLI)	410
	Log File Options	410
	Event Levels	411
	Log Groups	412
	Overview of SNMP Support for the SIC (SRC CLI)	413
Chapter 32	Configuring the Subscriber Information Collector with the SRC CLI	415
	Configuring the Connection Between the SIC and the Juniper Networks Database (SRC CLI)	416
	Creating a SIC Group (SRC CLI)	417
	Configuring a Basic SIC Group (SRC CLI)	417
	Creating a SIC Server Instance (SRC CLI)	419
	Adding a Server to a SIC Group (SRC CLI)	419
	Configuring the SSR Database as the Accounting Method (SRC CLI)	420
	Configuring Proxy RADIUS as the Accounting Method (SRC CLI)	421
	Configuring the RADIUS Accounting Listener for the SIC Group (SRC CLI)	422
	Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI)	423
	Configuring the RADIUS Transport for a SIC Server (SRC CLI)	425
	Configuring Dictionaries for the SIC Group (SRC CLI)	426
	Loading a Dictionary into the SIC Shared Group Configuration (SRC CLI)	428
	Configuring the Device Models Supported by the SIC Group (SRC CLI)	429
	Configuring Upstream RADIUS Network Elements and Clients for the SIC Group (SRC CLI)	430
	Configuring Downstream RADIUS Network Elements and Accounting Targets for the SIC Group (SRC CLI)	431
	Configuration Statements for Downstream Network Elements and Accounting Targets	431
	Configuring the Downstream Network Element and Device Model	432
	Configuring SIC Accounting Targets (SRC CLI)	433
	Configuring the Failover Policy for the Network Element (SRC CLI)	433
	Configuring the Fast Fail Options for the Failover Policy	434
	Configuring the Retry Options for the Failover Policy	434

	Configuring What Realms Are Local to the SIC Group (SRC CLI)	435
	Configuration Statements for SIC Editing Rules (SRC CLI)	435
	Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI)	439
	Configuration Statements for SIC Explicit Accounting Routing Rules	441
	Configuring Explicit Accounting Routes for the SIC (SRC CLI)	442
	Configuring Implicit Accounting Routes for the SIC (SRC CLI)	444
	Configuring Event Logging for a SIC Server (SRC CLI)	445
	Configuring SNMP for the SIC Group (SRC CLI)	448
	Example: Basic SIC Group Configuration (SRC CLI)	449
Chapter 33	Monitoring the Subscriber Information Collector with the SRC CLI	461
	Viewing Statistics for Accounting Routes (SRC CLI)	461
	Viewing Statistics for RADIUS Client Accounting Requests (SRC CLI)	461
	Viewing RADIUS Host Statistics for Accounting Transactions (SRC CLI)	461
	Viewing RADIUS Target Statistics for Accounting Requests (SRC CLI)	462
Part 9	Index	
	Index	465

List of Figures

Part 1	Operating the SAE	
Chapter 1	Overview of the SAE	3
	Figure 1: SAE Plug-In Architecture	6
	Figure 2: SRC SAE APIs	8
	Figure 3: Remote Interface on the SAE	9
	Figure 4: Sending Accounting Data to a RADIUS Server	10
	Figure 5: Sending Accounting Data to an Accounting File	10
	Figure 6: Customer Choice for SRC Accounting Deployment	10
Part 3	Using Network Devices in the SRC Network	
Chapter 8	Integrating Third-Party Network Devices into the SRC Network (SRC CLI)	91
	Figure 7: SAE Community	92
	Figure 8: Login Interactions with Assigned IP Subscribers	94
	Figure 9: Login Interactions with Event Notification Application	95
Part 4	Locating Subscriber Management Information	
Chapter 9	Locating Subscriber Information with the NIC	111
	Figure 10: Communication Between a NIC Proxy and a NIC Host in Client/Server Mode	112
	Figure 11: Communication Between a NIC Host and a NIC Proxy in Local Host Mode	112
	Figure 12: NIC Groups	115
	Figure 13: NIC Group Selection by Round-Robin	116
	Figure 14: NIC Resolution Request	117
Chapter 16	NIC Configuration Scenarios	187
	Figure 15: Resolution Process for ip Realm	188
	Figure 16: OnePop Centralized Configuration	189
	Figure 17: OnePop Distributed Configuration	189
	Figure 18: Redundancy for OnePop Centralized Configuration	190
	Figure 19: Resolution Process for Pcmam Realm	191
	Figure 20: OnePopPcmam Centralized Configuration	192
	Figure 21: OnePopPcmam Distributed Configuration	192
	Figure 22: Resolution Process for dynamicip Realm	193
	Figure 23: OnePopDynamicip Centralized Configuration	194
	Figure 24: OnePopDynamicip Distributed Configuration	194
	Figure 25: Resolution Process for sharedip Realm	195
	Figure 26: OnePopSharedIP Centralized Configuration	196

Figure 27: OnePopSharedIP Distributed Configuration	196
Figure 28: Resolution Process for the StaticRouteIp Realm	197
Figure 29: OnePopStaticRouteIp Centralized Configuration	198
Figure 30: OnePopStaticRouteIp Distributed Configuration	198
Figure 31: Resolution Process for the VrfIp Realm	199
Figure 32: OnePopVrfIp Centralized Configuration	200
Figure 33: OnePopStaticRouteIp Distributed Configuration	201
Figure 34: Resolution Process for acctId Realm	201
Figure 35: OnePopAcctId Centralized Configuration	203
Figure 36: Resolution Processes login Realm	203
Figure 37: OnePopLogin Centralized Configuration	205
Figure 38: OnePopLogin Distributed Configuration	205
Figure 39: OnePopLoginPull Distributed Configuration	206
Figure 40: Resolution Processes for primary_user Realm	206
Figure 41: OnePopPrimaryUser Centralized Configuration	207
Figure 42: OnePopPrimaryUser Distributed Configuration	208
Figure 43: OnePopDnSharedIp Realms Centralized Configuration	209
Figure 44: OnePopDnSharedIp Realms Distributed Configuration	211
Figure 45: OnePopAllRealms Centralized Configuration	213
Figure 46: OnePopAllRealms Distributed Configuration	215
Figure 47: MultiPop Configuration	217
Figure 48: iP Realm for MultiPop Configuration	218
Figure 49: sharedIP Realm for MultiPop Configuration	219
Figure 50: Resolution Graph for MultiPOP dn Realm	219
Figure 51: dn Realm for MultiPop Configuration	220

Part 5

Chapter 17

Providing Admission Control with SRC ACP

Overview of Providing Admission Control with SRC ACP 223

Figure 52: Position of SRC ACP in Network 224

Part 7

Chapter 26

Using Session State Registrar

Session State Registrar Overview 333

Figure 53: SSR with Four Data Nodes in Two Groups 336

Figure 54: Basic Session State Registrar Cluster 337

Figure 55: SSR Cluster with Four Data Nodes Forming Two-Node Groups 339

Figure 56: SSR Cluster with Redundant Network 340

Figure 57: SSR Cluster with Redundant Network 343

Figure 58: SSR Cluster Divided Between Two Sites with Tertiary Management
Server 345

Figure 59: SSR Cluster Evenly Divided Between Two Sites 346

Chapter 27

Planning Your Session State Registrar Cluster 355

Figure 60: Basic SSR Starter Kit Cluster 355

Part 8

Chapter 31

Using the Subscriber Information Collector

Overview of the Subscriber Information Collector 393

Figure 61: Configuration of SIC Servers and Groups 395

Figure 62: Explicit Routing Rule Process	400
Figure 63: SIC Editing Rule Process	402
Figure 64: Editing and Accounting Routing Rule Conditions and Processes . . .	403
Figure 65: Upstream and Downstream Network Elements	405
Figure 66: SNMP Support for the SIC	413

List of Tables

	About the Documentation	xxix
	Table 1: Notice Icons	xxx
	Table 2: Text Conventions	xxx
Part 2	Using Juniper Networks Routers in the SRC Network	
Chapter 6	Using JUNOSe Routers in the SRC Network (SRC CLI)	45
	Table 3: Router Initialization Scripts	53
	Table 4: Exported Fields	54
Part 3	Using Network Devices in the SRC Network	
Chapter 8	Integrating Third-Party Network Devices into the SRC Network (SRC CLI)	91
	Table 5: Router Initialization Scripts	104
	Table 6: Exported Fields	104
Part 4	Locating Subscriber Management Information	
Chapter 9	Locating Subscriber Information with the NIC	111
	Table 7: Types of NIC Agents	114
	Table 8: NIC Configuration Scenarios	118
	Table 9: NIC Agents	122
	Table 10: Agents in Configuration Scenarios	123
	Table 11: Type of Router Initialization Script to Use for NIC Configuration Scenarios	124
Chapter 15	NIC Resolution Process	181
	Table 12: Available NIC Resolutions	181
Part 5	Providing Admission Control with SRC ACP	
Chapter 17	Overview of Providing Admission Control with SRC ACP	223
	Table 13: Congestion Points Derived Through NAS Port ID	226
Part 6	Using External Subscriber Monitor	
Chapter 24	Monitoring External Subscriber Events with the SRC CLI	325
	Table 14: Output Fields for show external-subscriber-monitor statistics radius-accounting	325
	Table 15: Output Fields for show external-subscriber-monitor statistics event-notifications	327

	Table 16: Output Fields for show external-subscriber-monitor statistics process	328
Part 7	Using Session State Registrar	
Chapter 26	Session State Registrar Overview	333
	Table 17: Latency Between Servers and Its Effect on Performance	340
	Table 18: Supported Cluster Configurations	341
	Table 19: Subscriber Sessions Table Default Fields	347
	Table 20: Service Sessions Table Default Fields	350
Chapter 27	Planning Your Session State Registrar Cluster	355
	Table 21: Example Allocation of Node IDs	356
	Table 22: SSR Starter Kit Cluster Worksheet	357
	Table 23: Expanded SSR Cluster Planning Worksheet	358
Part 8	Using the Subscriber Information Collector	
Chapter 31	Overview of the Subscriber Information Collector	393
	Table 24: Example of SSR Database Mapping	398
	Table 25: SIC Log File Options	410
	Table 26: SIC Event Levels	412
	Table 27: SIC Log Groups	412
	Table 28: MIBs Used by the SIC for SNMP Statistics	414
	Table 29: SNMP Traps Supported for the SIC	414
Chapter 32	Configuring the Subscriber Information Collector with the SRC CLI	415
	Table 30: SIC Editing Rule Options	439
	Table 31: Explicit Routing Rule Conditions	443
	Table 32: Sample Configuration Attribute Associations	449
	Table 33: Log Groups and Associated Event Level for Log Stream=default logger	450
	Table 34: Log Groups and Associated Event Level for Log Stream=debug-logger	450
	Table 35: Log Groups and Associated Event Level for Log Stream=error-logger	451

About the Documentation

- SRC Documentation and Release Notes on page xxix
- Audience on page xxix
- Documentation Conventions on page xxix
- Documentation Feedback on page xxxi
- Requesting Technical Support on page xxxi

SRC Documentation and Release Notes

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

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

Audience

This documentation is intended for experienced system and network specialists working with routers running JUNOS® 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 xxx defines the notice icons used in this guide. Table 2 on page xxx defines text conventions used throughout this documentation.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2: Text Conventions

Convention	Description	Examples
Bold text like this	<ul style="list-style-type: none"> Represents keywords, scripts, and tools in text. Represents a GUI element that the user selects, clicks, checks, or clears. 	<ul style="list-style-type: none"> Specify the keyword exp-msg. Run the install.sh script. Use the pkgadd tool. To cancel the configuration, click Cancel.
Bold text like this	Represents text that the user must type.	user@host# set cache-entry-age cache-entry-age
Fixed-width text like this	Represents information as displayed on your terminal's screen, such as CLI commands in output displays.	<pre>nic-locators { login { resolution { resolver-name /realms/ login/A1; key-type LoginName; value-type SaeId; } } }</pre>
Regular sans serif typeface	<ul style="list-style-type: none"> Represents configuration statements. Indicates SRC CLI commands and options in text. Represents examples in procedures. Represents URLs. 	<ul style="list-style-type: none"> system ldap server{ stand-alone; Use the request sae modify device failover command with the force option user@host# ... http://www.juniper.net/techpubs/software/ management/src/api-index.html
<i>Italic sans serif typeface</i>	Represents variables in SRC CLI commands.	user@host# set local-address local-address
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <gfwif>.
Key name	Indicates the name of a key on the keyboard.	Press Enter.

Table 2: Text Conventions (*continued*)

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

Documentation Feedback

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

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

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

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

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

Opening a Case with JTAC

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

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

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

PART 1

Operating the SAE

- Overview of the SAE on page 3
- Configuring the SAE (SRC CLI) on page 13
- Managing Subscriber and Service Session Data (SRC CLI) on page 23
- Managing SAE Data (SRC CLI) on page 31
- Managing SAE Data (C-Web Interface) on page 37

CHAPTER 1

Overview of the SAE

This chapter gives an overview of the features of the SAE. Topics include:

- Role of the SAE on page 3
- Connections to Managed Devices on page 3
- SAE Plug-Ins on page 5
- Tracking and Controlling Subscriber and Service Sessions with SAE APIs on page 7
- SAE Accounting on page 9

Role of the SAE

The SAE is the core manager of the SRC network. It interacts with other systems, such as Juniper Networks routers, cable modem termination system (CMTS) devices, directories, Web application servers, and RADIUS servers, to retrieve and disseminate data in the SRC environment. The SAE authorizes, activates and deactivates, and tracks subscriber and service sessions. It also collects accounting information about subscribers and services.

The SAE makes decisions about the deployment of policies on JUNOSe routers and JUNOS routing platforms. When a subscriber's IP interface comes up on the router, the SAE determines whether it manages the interface. If the interface is managed—or controlled—by the SAE, the SAE sends the subscriber's default policy configuration to the router. These default policies define the subscriber's initial network access. When the subscriber activates a value-added service, the SAE translates the service into lists of policies and sends them to the router.

The SAE also provides plug-ins and application programming interfaces (APIs) that extend the capabilities of the SRC software.

- Related Topics**
- Tracking and Controlling Subscriber and Service Sessions with SAE APIs on page 7
 - Connections to Managed Devices on page 3

Connections to Managed Devices

This topic describes the connections between the SAE and Juniper Networks routers, CMTS devices, and the Juniper Policy Server (JPS).

COPS Connection Between JUNOSe Routers and the SAE

The SAE and JUNOSe routers communicate using the Common Open Policy Service (COPS) protocol. The SAE supports two versions of COPS:

- COPS usage for policy provisioning (COPS-PR)
- COPS External Data Representation Standard (XDR) mode

The version of COPS that you use depends on the version of COPS that your JUNOSe router supports. When you set up your JUNOSe router to work with the SAE, you enable either COPS-PR mode or COPS XDR mode. There are no configuration differences on the SAE between COPS-PR and COPS XDR.

The following SRC features require the use of COPS-PR:

- Policy sharing on JUNOSe routers
- Multiple classify traffic conditions in policy lists

Beep Connection Between JUNOS Routing Platforms and the SAE

The SAE interacts with a JUNOS software process, referred to as the SRC software process, on a JUNOS routing platform. The SAE and the SRC software process communicate using the Blocks Extensible Exchange Protocol (BEEP).

When a JUNOS routing platform that the SAE manages goes online, it initiates a BEEP session for the SAE. The SAE gets configuration information from the router, and then it builds and installs the policies that control the router's behavior. If the policies are subsequently modified in the directory, the SAE builds a new configuration and reconfigures the interface on the JUNOS routing platform.



NOTE: The SAE manages interfaces on JUNOS routing platforms only when the interfaces are configured in the global configuration and the router sends added, changed, or deleted notifications to the SAE. Router administrators should not manually change the configuration of interfaces that the SAE is managing. If you manually change a configuration, you must remove the SAE from the system.

When there are configuration changes on the router, the router sends a notification to the SAE through the BEEP connection. The notification does not include the content of the configuration changes. When the SAE receives the notification, it uses its JUNOScript client to get the changed configuration from the router.

Interfaces that have been deleted from the router along with their associated objects (sessions, policies) remain on the router until state synchronization occurs.

COPS Connection Between CMTS Devices and the SAE

The SAE uses the COPS protocol as specified in the [PacketCable Multimedia Specification PKT-SP-MM-103-051221](#) to manage *PacketCable Multimedia Specification* (PCMM)-compliant CMTS devices in a cable network environment. The SAE connects

to the CMTS device by using a COPS over Transmission Control Protocol (TCP) connection.

In cable environments, the SAE manages the connection to the CMTS device. The CMTS device does not provide address requests or notify the SAE of new subscribers, subscriber IP addresses, or any other attributes. IP address detection and all other subscriber attributes are collected outside of the COPS connection to the CMTS device. The SAE uses COPS only to push policies to the CMTS device and to learn about the CMTS status and usage data.

Because the CMTS device does not have the concept of interfaces, the SRC module uses pseudointerfaces to model CMTS subscriber connections similar to subscriber connections for JUNOS routing platforms and JUNOSe routers.

COPS Connection Between Juniper Policy Servers and the SAE

When the SAE is acting as an application manager in a PCMM environment, it connects to the JPS through an interface on the JPS. The JPS uses the COPS protocol as specified in the PacketCable Multimedia Specification PKT-SP-MM-I03-051221 for its interface connections. The JPS communicates with the application manager by using a COPS over TCP connection.

For more information, see .

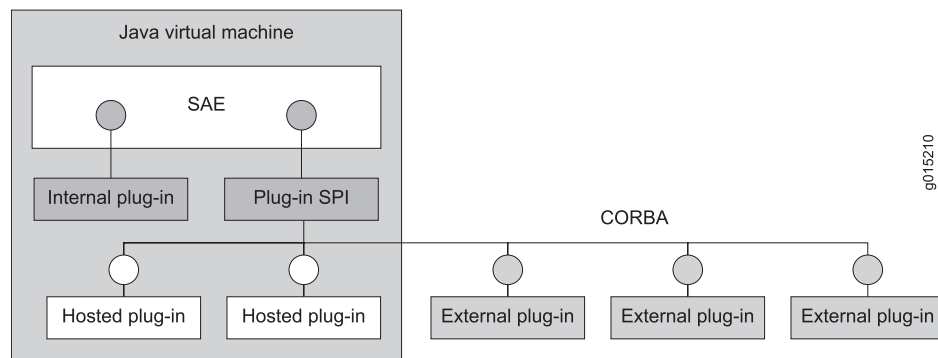
- Related Topics**
- Overview of a PCMM Environment
 - Overview of the JPS
 - Configuring the SAE to Manage JUNOS Routing Platforms (SRC CLI) on page 69
 - Adding JUNOSe Routers and Virtual Routers (SRC CLI) on page 46

SAE Plug-Ins

Plug-ins are software programs that extend the capabilities of existing programs and make them more flexible. SRC plug-ins provide authentication, authorization, and tracking capabilities.

There are three types of plug-ins: internal, hosted, and external. Internal plug-ins communicate directly with the SAE. Hosted and external plug-ins implement a published Common Object Request Broker Architecture (CORBA)-based service provider interface (SPI), which means that anyone with access to the interface specification can create plug-ins that work with the SRC module. Figure 1 on page 6 gives an overview of the plug-in architecture.

Figure 1: SAE Plug-In Architecture



Internal Plug-Ins

The SRC module provides internal plug-ins that perform a range of authentication, authorization, and tracking functions. With these plug-ins, you can, for example, authenticate subscribers, authorize subscriptions and sessions, authorize IP address requests from DHCP clients, track subscriber activity and service use, track quality of service (QoS) services and attach and remove QoS profiles as needed, and limit the number of authenticated subscribers who connect to an IP interface on the router.

Internal plug-ins implement an interface that communicates directly with the SAE. They have the following characteristics:

- Run within the SAE's Java Virtual Machine (JVM)
- Are started and stopped with the SAE
- Are implemented in Java

The core SRC module provides a set of internal plug-ins. .

External Plug-Ins

The SRC module includes the SAE CORBA plug-in SPI. This SPI allows you to implement external plug-ins in any language that supports CORBA (for example, Java, C++, Python), which makes it easy to integrate the SAE with operations support system (OSS) software written in a wide variety of languages and distributed across a variety of hardware and operating system platforms.

External plug-ins link a service provider's OSS with the SAE so that the OSS is notified of events in the life cycle of SAE sessions. For example, plug-ins can be notified when a subscriber attempts to log in and begins the authentication and authorization process. This notification makes it possible for the plug-in to consult general data and resource allocation information that is available to the OSS, and use that information to make authorization decisions.

The SPI also sends session-tracking events when sessions start, on an interim basis, and when sessions stop. Plug-ins can set session timeouts as a response to both session start and interim events. This capability enables the development of prepaid applications

where the plug-in consults the subscriber's current account balance before it makes the decision to extend or reduce a session timeout.

External plug-ins have the following characteristics:

- Run outside the SAE's JVM, either in the same or in a different server
- Are implemented in any language that supports CORBA
- Communicate with the SAE using CORBA
- Support the admission control or prepaid demo plug-in, which can be purchased separately from the SRC module.

Hosted Plug-Ins

Hosted plug-ins, like the external ones, implement the CORBA interface. Unlike the external ones, hosted plug-ins are instantiated (that is, hosted) by the SAE. As a result, they live in the same JVM process as the host SAE, which means that hosted plug-ins must be implemented in Java.

Hosted plug-ins have the following characteristics:

- Run within the SAE's JVM
- Communicate with SAE using CORBA
- Are started and stopped with SAE
- Are implemented using a published interface

- Related Topics**
- How Internal Plug-Ins Work
 - Connections to Managed Devices on page 3
 - Configuring the SAE for External Plug-Ins (SRC CLI)
 - The interface definition language (IDL) code and online documentation for the SAE CORBA Plug-In SPI is on the Juniper Networks Web site at <https://www.juniper.net/support/csc/swdist-erx/src.html>.

Tracking and Controlling Subscriber and Service Sessions with SAE APIs

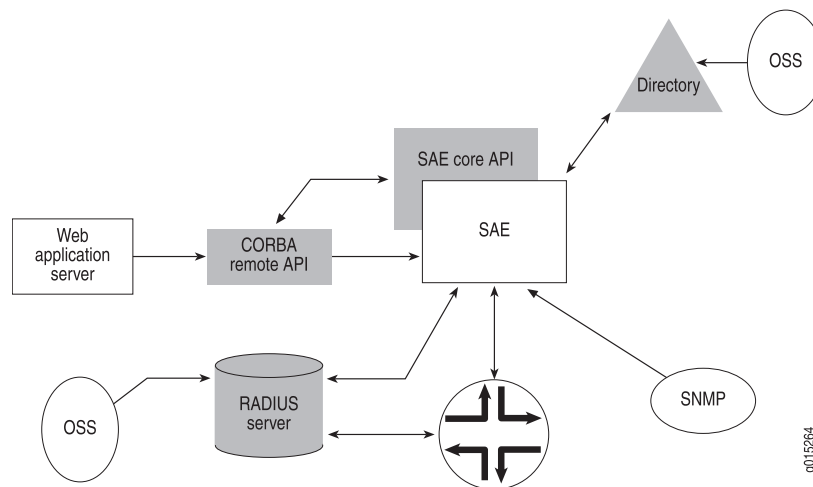
The SAE provides two public APIs:

- SAE core API
- SAE CORBA remote API

Through these interfaces, an external application can track and control subscriber and service sessions.

Figure 2 on page 8 illustrates the SAE APIs.

Figure 2: SRC SAE APIs



SAE Core API

The SAE core API is used to control the behavior of the SRC module. There are many uses of the SAE core API. For example, it can be used to provide:

- Subscriber credentials (username/password)
- Requests for service activation/deactivation for a subscriber

This API can be used by a Java application running in the same JVM as the SAE. For example, you can access the SAE core API from plug-ins that are hosted by the SAE, or you can use the SAE core API to write your own extensions of the SAE remote interface by using CORBA or the SAE script interface modules.

SAE CORBA Remote API

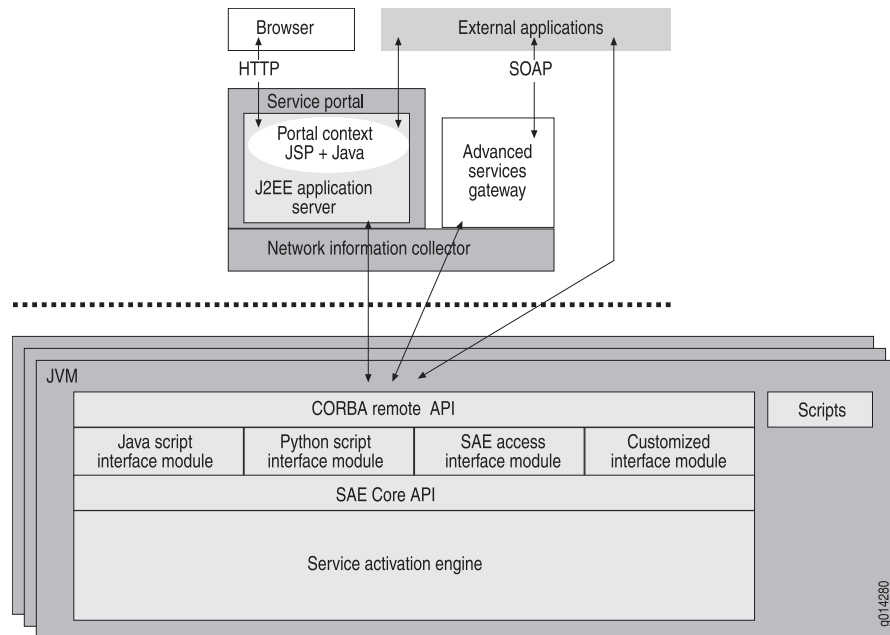
This API provides a way to use external applications with the SRC module (see Figure 3 on page 9). All functions that are available through the SAE core API are available through the CORBA remote API. The remote API provides several remote interfaces that allow customization of the API for special needs. The remote interface comprises an interface module manager and a set of interface modules. We provide the following interface modules with the SRC module:

- SAE access interface module—Provides remote access to the SAE core API
- Java script interface module—Allows you to control the SAE with a Java script
- Python script interface module—Allows you to control the SAE with a Python script
- Event notification interface module—Allows you to integrate the SAE with external IP address managers

You can also create custom interface modules that allow external applications to extend the capabilities of the SAE. To do so, you must define the interface module in CORBA IDL and implement it in Java.

The remote interface publishes one object reference that acts as the interface module manager. External applications communicate through CORBA with the interface module manager to retrieve a particular interface module. That interface module runs in the same JVM as the SAE and has full access to the SAE core API.

Figure 3: Remote Interface on the SAE



For more information about the SAE CORBA remote API, including the interfaces, properties, and methods, see the online documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.

- Related Topics**
- Storing Subscriber and Service Session Data on page 23
 - Configuring Access to Subscriber Data (SRC CLI) on page 15
 - Configuring Access to Service Data (SRC CLI) on page 17
 - Configuring Access Through LDAPS to Service and Subscriber Data (SRC CLI) on page 14

SAE Accounting

The router and the SAE generate RADIUS accounting records when subscribers access the Internet and use value-added services. The records are sent to RADIUS accounting servers and are logged in accounting log files, or they are sent to accounting flat files. External systems collect the accounting log files and feed them to a rating and billing system.

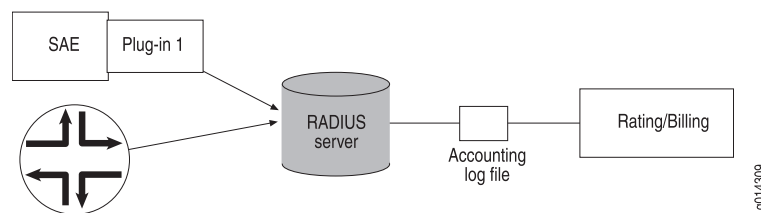
The SRC module allows a variety of accounting deployments. This topic shows the standard deployment that we supply, a second option that does not depend on a RADIUS

server, and a third option in which customers develop their own deployment by choosing a CORBA plug-in.

In the standard SRC deployment (see Figure 4 on page 10), the router and the SAE are clients of the RADIUS accounting server. They pass subscriber accounting information to a designated RADIUS accounting server in an accounting request. The RADIUS accounting server receives the accounting request and creates accounting log files.

The SRC module works with other AAA RADIUS servers; however, we validate the SRC module only with Merit, Interlink RAD-Series AAA RADIUS Server, or Juniper Networks Steel-Belted Radius/SPE server.

Figure 4: Sending Accounting Data to a RADIUS Server



A second option, shown in Figure 5 on page 10, uses an accounting flat file generated directly by the SAE, without a RADIUS server.

Figure 5: Sending Accounting Data to an Accounting File

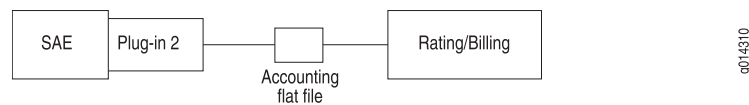
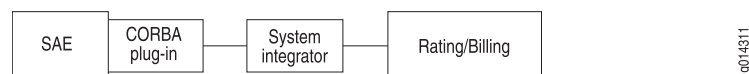


Figure 6 on page 10 illustrates a third possibility, one in which the customer uses a CORBA plug-in of his or her own choice.

Figure 6: Customer Choice for SRC Accounting Deployment



Accounting Policy

The SAE defines the policies that control the network traffic for the subscriber based on the subscriber's subscriptions. It also determines the accounting statistics collected for the subscribed service.

While defining the policies for a service, the SAE can choose the policy rules to be used for accounting per interface direction (ingress and egress). Statistics are collected for the chosen policy rules for the service and are sent to the RADIUS accounting server. The SAE can also decide not to collect any policy rule-specific statistics for the service. In this case, only session times are sent to the accounting system when the service is deactivated. When choosing multiple policy rules on traffic direction for statistics collection, the SAE summarizes the statistics by adding the individual values.

Subscription Process

After an outsourced service has been set up, subscribers can order primary access or value-added services from retailers, who in turn notify the wholesaler of the new end subscription. Conversely, accounting data is collected by the wholesaler and communicated to the retailer to provide enough data for the retailer to bill the subscriber.

The overall subscription process is simplified:

- The subscriber has no need to interact with another party or a device other than the router.
- When the subscriber goes to the Web portal and selects the service, the subscription activation is triggered.
- The subscriber's portal page adjusts to display the new service.
- Accounting data is generated, identifying the service being tracked for the subscriber.

Tracking Subscriber Sessions

The intelligent service accounting function of the SRC module tracks the subscription activity for each subscriber and each service session. It collects usage information and passes the information to the appropriate rating and billing system.

Multiple service sessions can be activated simultaneously for a subscriber and can be tracked separately from an accounting standpoint.

Events are generated when service sessions are activated and deactivated, and during interim accounting updates.

Accounting Plug-Ins

Plug-ins allow service providers to easily extend the capabilities of their systems through the use of plug-in software. See "SAE Plug-Ins" on page 5.

Interim Accounting

The router and SAE generate interim accounting records for broadband primary services (through PPP) and value-added services, respectively. RADIUS servers log the interim records in their accounting log files when interim accounting is enabled.

The external rating system calculates the charges by using interim records instead of stop records for timeout sessions. The calculation occurs when the last record is interim and for open sessions whose last record at the end of a billing cycle is interim.

An accounting interim interval is defined for each service and applied to all subscriptions to that service. The router and SAE generate accounting requests with a status of interim for every period of time specified with the interim value.

The router receives an accounting interim value for a session through a RADIUS server when the router makes an authentication request. If the RADIUS server does not provide a value, then the router does not generate interim accounting records.

The SAE obtains an accounting interim value from the directory. When the accounting interim value is not stored, the SAE uses global values. When a value equals zero, the SAE does not generate interim accounting records.

- Related Topics**
- Role of the SAE on page 3
 - Connections to Managed Devices on page 3
 - SAE Plug-Ins on page 5
 - Tracking and Controlling Subscriber and Service Sessions with SAE APIs on page 7

CHAPTER 2

Configuring the SAE (SRC CLI)

- SRC Access to Directory Data on page 13
- Configuring LDAP Access to Directory Data (SRC CLI) on page 14
- Configuring Access Through LDAPS to Service and Subscriber Data (SRC CLI) on page 14
- Configuring Access to Subscriber Data (SRC CLI) on page 15
- Configuring Access to Service Data (SRC CLI) on page 17
- Configuring Access to Policy Data (SRC CLI) on page 18
- Configuring Access to the Persistent Login Cache (SRC CLI) on page 19
- Configuring the Location of Network Device Data (SRC CLI) on page 20
- Enabling Automatic Discovery of Changes in SAE Configuration Data (SRC CLI) on page 21
- Setting the Timeout and Number of Events for SAE Directory Eventing (SRC CLI) on page 22

SRC Access to Directory Data

The SRC module stores subscriber, service, persistent login, policy, router, and cached subscriber profiles and session data in a directory. The SAE uses LDAP to store and retrieve the data.

If you do not store data in the local directory, you need to configure the LDAP connections to the directories in which the data is stored. You can also select the filter that the SAE uses to search for subscriptions in the directory and directory eventing parameters for data stored in the directory.

Related Topics

- Storing Subscriber and Service Session Data on page 23
- Configuring LDAP Access to Directory Data (SRC CLI) on page 14
- Configuring Access to the Persistent Login Cache (SRC CLI) on page 19
- Configuring Access to Policy Data (SRC CLI) on page 18
- Configuring Access Through LDAPS to Service and Subscriber Data (SRC CLI) on page 14

Configuring LDAP Access to Directory Data (SRC CLI)

The tasks to configure LDAP access to directory data are:

- (Optional) “Configuring Access Through LDAPS to Service and Subscriber Data (SRC CLI)” on page 14
- Configuring Access to Subscriber Data (SRC CLI) on page 15
- Configuring Access to Service Data (SRC CLI) on page 17
- Configuring Access to Policy Data (SRC CLI) on page 18
- Configuring Access to the Persistent Login Cache (SRC CLI) on page 19
- Configuring the Location of Network Device Data (SRC CLI) on page 20
- Enabling Automatic Discovery of Changes in SAE Configuration Data (SRC CLI) on page 21
- Setting the Timeout and Number of Events for SAE Directory Eventing (SRC CLI) on page 22

- Related Topics**
- Configuring LDAP Access to Directory Data (C-Web Interface)
 - Storing Subscriber and Service Session Data on page 23
 - SRC Access to Directory Data on page 13
 - Connections to Managed Devices on page 3

Configuring Access Through LDAPS to Service and Subscriber Data (SRC CLI)

You can secure connections between a router and an external directory that contains service data or subscriber data, and you can configure the router to use LDAPS when it connects to the same data source.

Use the following configuration statements to configure access through LDAPS to service data and subscriber data:

```
shared sae configuration ldap service-data {  
    (ldaps);  
}  
  
shared sae configuration ldap subscriber-data {  
    (ldaps);  
}
```

To use LDAPS to secure connections between a router and an external directory:

1. Configure the directory connection from the SAE to use LDAPS. For example:

```
user@host# set shared sae configuration ldap service-data ldaps  
user@host# set shared sae configuration ldap subscriber-data ldaps
```
2. In the router initialization script you specify the directory context.

The `/opt/UMC/sae/lib/poolPublisher.py` script and the `/opt/UMC/sae/lib/lorPublisher.py` script provide examples of how to configure a directory context. For example, from the `/opt/UMC/sae/lib/lorPublisher.py` script:

```
dirContext = Ssp.registry.get('ServiceDataSource.component').getContext()
```

In addition, you can change the directory context.

For information about how to use `InitialDirContext` class or the `DirContext` class to specify directory context, see:

<http://java.sun.com/j2se/1.4.2/docs/api/javax/naming/directory/InitialDirContext.html>

<http://java.sun.com/j2se/1.4.2/docs/api/javax/naming/directory/DirContext.html>

- Related Topics**
- Configuring Access to Subscriber Data (SRC CLI) on page 15
 - Configuring Access Through LDAPS to Service and Subscriber Data (C-Web Interface)
 - Configuring Access to Service Data (SRC CLI) on page 17
 - Configuring Access to Policy Data (C-Web Interface)
 - SRC Access to Directory Data on page 13

Configuring Access to Subscriber Data (SRC CLI)

Use the following configuration statements to configure access to subscriber data:

```
shared sae configuration ldap subscriber-data {
  subscription-loading-filter (subscriberRefFilter | objectClassFilter);
  load-subscriber-schedules;
  login-cache-dn login-cache-dn ;
  session-cache-dn session-cache-dn ;
  server-address server-address ;
  dn dn ;
  authentication-dn authentication-dn ;
  password password ;
  directory-eventing;
  polling-interval polling-interval ;
  (ldaps);
}
```

To configure SAE access to subscriber data:

1. From configuration mode, access the configuration statement that configures SAE access to subscriber data in the directory. In this sample procedure, the subscriber data is configured in the `se-region` group.

```
user@host# edit shared sae group se-region configuration ldap subscriber-data
```

2. Select the filter that the SAE uses to search for subscriptions in the directory when the SAE loads a subscription to a subscriber reference filter.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set subscription-loading-filter (subscriberRefFilter | objectClassFilter)
```

3. (Optional) Enable loading of subscriber schedules.

- [edit shared sae group se-region configuration ldap subscriber-data]
user@host# **set load-subscriber-schedules**
4. Specify the subtree in the directory in which subscriber information is stored.
- [edit shared sae group se-region configuration ldap subscriber-data]
user@host# **set login-cache-dn *login-cache-dn***
5. Specify the subtree in the directory in which persistent session data is cached.
- [edit shared sae group se-region configuration ldap subscriber-data]
user@host# **set session-cache-dn *session-cache-dn***
6. (Optional) Specify the directory server that stores subscriber information.
- [edit shared sae group se-region configuration ldap subscriber-data]
user@host# **set server-address *server-address***
7. Specify the subtree in the directory where subscriber data is cached.
- [edit shared sae group se-region configuration ldap subscriber-data]
user@host# **set dn *dn***
8. (Optional) Specify the DN that the SAE uses to authenticate access to the directory server.
- [edit shared sae group se-region configuration ldap subscriber-data]
user@host# **set authentication-dn *authentication-dn***
9. (Optional) Specify the password used to authenticate access to the directory server.
- [edit shared sae group se-region configuration ldap subscriber-data]
user@host# **set password *password***
10. (Optional) Enable automatic discovery of changes in subscriber profiles.
- [edit shared sae group se-region configuration ldap subscriber-data]
user@host# **set directory-eventing**
11. Set the frequency for checking the directory for updates.
- [edit shared sae group se-region configuration ldap subscriber-data]
user@host# **set polling-interval *polling-interval***
12. Enable LDAPS as the secure protocol for connections to the server that stores subscriber data.
- [edit shared sae group se-region configuration ldap subscriber-data]
user@host# **set ldaps**
13. (Optional) Verify your configuration.
- [edit shared sae group se-region configuration ldap subscriber-data]
user@host# **show**
subscription-loading-filter objectClassFilter;
load-subscriber-schedules;
login-cache-dn o=users,<base>;
session-cache-dn o=PersistentSessions,<base>;
server-address 127.0.0.1;
dn o=users,<base>;
authentication-dn cn=ssp,o=components,o=operators,<base>;
password *****;
directory-eventing;


```
polling-interval 30;
ldaps;
```

- Related Topics**
- Creating Grouped Configurations for the SAE (SRC CLI)
 - Configuring Access Through LDAPS to Service and Subscriber Data (SRC CLI) on page 14
 - Configuring Access to Service Data (SRC CLI) on page 17
 - Viewing General Information for Subscriber Sessions (SRC CLI)
 - Viewing Statistics for Subscriber and Service Sessions (SRC CLI)

Configuring Access to Service Data (SRC CLI)

Use the following configuration statements to configure access to service data:

```
shared sae configuration ldap service-data {
  server-address server-address;
  dn dn;
  authentication-dn authentication-dn;
  password password;
  directory-eventing;
  polling-interval polling-interval;
  (ldaps);
}
```

To configure SAE access to service data:

1. From configuration mode, access the configuration statement that configures SAE access to service data in the directory. In this sample procedure, the service data is configured in the se-region group.


```
user@host# edit shared sae group se-region configuration ldap service-data
```
2. (Optional) Specify the directory server that stores service data.


```
[edit shared sae group se-region configuration ldap service-data]
user@host# set server-address server-address
```
3. Specify the subtree in the directory where service data is cached.


```
[edit shared sae group se-region configuration ldap service-data]
user@host# set dn dn
```
4. (Optional) Specify the DN that the SAE uses to authenticate access to the directory server.


```
[edit shared sae group se-region configuration ldap service-data]
user@host# set authentication-dn authentication-dn
```
5. (Optional) Specify the password used to authenticate access to the directory server.


```
[edit shared sae group se-region configuration ldap service-data]
user@host# set password password
```
6. (Optional) Enable or disable automatic discovery of changes to service data.

```
[edit shared sae group se-region configuration ldap service-data]
user@host# set directory-eventing
```

7. Set the frequency for checking the directory for updates.

```
[edit shared sae group se-region configuration ldap service-data]
user@host# set polling-interval polling-interval
```

8. Enable LDAPS as the secure protocol for connections to the server that stores service data.

```
edit shared sae group se-region configuration ldap service-data]
user@host# set ldaps
```

9. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration ldap service-data]
user@host# show
server-address 10.10.45.3;
dn <base>;
authentication-dn <base>;
password *****;
directory-eventing;
polling-interval 30;
ldaps;
```

- Related Topics**
- Creating Grouped Configurations for the SAE (SRC CLI)
 - Configuring Access to Subscriber Data (SRC CLI) on page 15
 - Configuring Access to Policy Data (SRC CLI) on page 18
 - Configuring Access Through LDAPS to Service and Subscriber Data (SRC CLI) on page 14

Configuring Access to Policy Data (SRC CLI)

Use the following configuration statements to configure access to policy data:

```
shared sae configuration ldap policy-data {
  policy-dn policy-dn ;
  parameter-dn parameter-dn ;
  directory-eventing;
  polling-interval polling-interval ;
}
```

To configure SAE access to subscriber data:

1. From configuration mode, access the configuration statement that configures SAE access to policy data in the directory. In this sample procedure, the policy data is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration ldap policy-data
```

2. Specify the subtree in the directory in which policy data stored.

```
[edit shared sae group se-region configuration ldap policy-data]
user@host# set policy-dn policy-dn
```

3. Specify the subtree in the directory in which policy parameter data is cached.

```
[edit shared sae group se-region configuration ldap policy-data]
user@host# set parameter-dn parameter-dn
```

4. (Optional) Enable or disable automatic discovery of changes to policy data.

```
[edit shared sae group se-region configuration ldap policy-data]
user@host# set directory-eventing
```

5. Set the frequency for checking the directory for updates.

```
[edit shared sae group se-region configuration ldap policy-data]
user@host# set polling-interval polling-interval
```

6. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration ldap policy-data]
user@host# show
policy-dn o=Policy,<base>;
parameter-dn o=Parameters,<base>;
directory-eventing;
polling-interval 30;
```

- Related Topics**
- Creating Grouped Configurations for the SAE (SRC CLI)
 - Configuring Access to Subscriber Data (SRC CLI) on page 15
 - Configuring Access to Service Data (SRC CLI) on page 17
 - Configuring Access to the Persistent Login Cache (SRC CLI) on page 19
 - SRC Access to Directory Data on page 13

Configuring Access to the Persistent Login Cache (SRC CLI)

Use the following configuration statements to configure access to persistent login cache data:

```
shared sae configuration ldap persistent-login-cache {
  server-address server-address;
  dn dn;
  authentication-dn authentication-dn;
  password password;
  directory-eventing;
  polling-interval polling-interval;
  (ldaps);
}
```

To configure SAE access to persistent login cache data:

1. From configuration mode, access the configuration statement that configures SAE access to persistent login cache data in the directory. In this sample procedure, the persistent login cache data is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration ldap persistent-login-cache
```

2. (Optional) Specify the directory server that stores service data.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set server-address server-address
```

3. Specify the subtree in the directory where persistent login cache data is cached.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set dn dn
```

4. (Optional) Specify the DN that the SAE uses to authenticate access to the directory server.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set authentication-dn authentication-dn
```

5. (Optional) Specify the password used to authenticate access to the directory server.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set password password
```

6. (Optional) Enable automatic discovery of changes to persistent login cache data.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set directory-eventing
```

7. Set the frequency for checking the directory for updates.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set polling-interval polling-interval
```

8. Enable LDAPS as the secure protocol for connections to the server that stores persistent login cache data.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set ldaps
```

9. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# show
dn "o=authCache, <base>";
directory-eventing;
polling-interval 30;
ldaps;
```

- Related Topics**
- Creating Grouped Configurations for the SAE (SRC CLI)
 - Configuring Access to Subscriber Data (SRC CLI) on page 15
 - Configuring Access to Service Data (SRC CLI) on page 17
 - Configuring Access to the Persistent Login Cache (C-Web Interface)
 - SRC Access to Directory Data on page 13

Configuring the Location of Network Device Data (SRC CLI)

Use the following configuration statement to configure access to network device data:

```
shared sae configuration ldap {
  network-dn network-dn ;
```

```
}
```

To configure SAE access to network device data:

1. From configuration mode, access the configuration statement that configures SAE access to network device data in the directory. In this sample procedure, the network device data is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration ldap
```

2. Specify the subtree in the directory where network device data is stored.

```
[edit shared sae group se-region configuration ldap]
```

```
user@host# set network-dn network-dn
```

3. Verify your configuration.

```
[edit shared sae group se-region configuration ldap]
```

```
user@host# show network-dn
```

```
network-dn o=Network,<base>;
```

- Related Topics**
- Creating Grouped Configurations for the SAE (SRC CLI)
 - Enabling Automatic Discovery of Changes in SAE Configuration Data (SRC CLI) on page 21
 - Configuring Access to the Persistent Login Cache (SRC CLI) on page 19
 - For more information about monitoring the SAE data with the SRC CLI, see Viewing Information About the Directory Blacklist (SRC CLI)

Enabling Automatic Discovery of Changes in SAE Configuration Data (SRC CLI)

Use the following configuration statement to enable automatic discovery of changes in SAE configuration data:

```
shared sae configuration ldap {
  enable-directory-eventing;
}
```

To enable automatic discovery of changes in SAE configuration data:

1. From configuration mode, access the configuration statement that enables automatic discovery of changes in SAE configuration data in the directory. In this sample procedure, automatic discovery is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration ldap
```

2. Enable automatic discovery of changes to SAE configuration data.

```
[edit shared sae group se-region configuration ldap]
```

```
user@host# enable-directory-eventing
```

- Related Topics**
- Creating Grouped Configurations for the SAE (SRC CLI)
 - Enabling Automatic Discovery of Changes in SAE Configuration Data (C-Web Interface)

- Setting the Timeout and Number of Events for SAE Directory Eventing (SRC CLI) on page 22

Setting the Timeout and Number of Events for SAE Directory Eventing (SRC CLI)

Use the following configuration statements to set the directory eventing timeout and the number of simultaneous events that the SAE can receive from the directory:

```
shared sae configuration ldap directory-eventing {  
    timeout timeout ;  
    dispatcher-pool-size dispatcher-pool-size ;  
}
```

To configure the directory eventing timeout and the number of simultaneous events that the SAE can receive from the directory:

1. From configuration mode, access the configuration statement that configures SAE directory eventing. In this sample procedure, directory eventing is configured in the `se-region` group.

```
user@host# edit shared sae group se-region configuration ldap directory-eventing
```

2. Specify the maximum time that the directory eventing system waits for the directory to respond.

```
[edit shared sae group se-region configuration ldap directory-eventing]  
user@host# set timeout timeout
```

3. Specify the number of events that the SAE can receive from the directory simultaneously.

```
[edit shared sae group se-region configuration ldap directory-eventing]  
user@host# set dispatcher-pool-size dispatcher-pool-size
```

4. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration ldap directory-eventing]  
user@host# show  
timeout 60;  
dispatcher-pool-size 1000;
```

Related Topics

- Creating Grouped Configurations for the SAE (SRC CLI)
- Setting the Timeout and Number of Events for SAE Directory Eventing (C-Web Interface)
- Enabling Automatic Discovery of Changes in SAE Configuration Data (SRC CLI) on page 21

CHAPTER 3

Managing Subscriber and Service Session Data (SRC CLI)

- Storing Subscriber and Service Session Data on page 23
- Configuring the Session Store Feature (SRC CLI) on page 25
- Configuring the Number of Threads for Sessions (SRC CLI) on page 29

Storing Subscriber and Service Session Data

To aid in recovering from an SAE failover, the SAE stores subscriber and service session data in flat files on the SAE host. The SRC component that controls the storage of session data on the SAE is called the session store. The session store queues data and then writes the data to session store files on the SAE host's disk. After the data has been written to disk, it can survive a server reboot.

You can configure how the SAE stores session data for JUNOSe routers, JUNOS routing platforms, simulated routers, and *PacketCable Multimedia Specification* (PCMM) devices.

Session Store Files

Session store files are numbered flat files. Session store files are located in a directory on the SAE host. You can configure the size of session store files. After the maximum size has been reached, the session store creates a new file and begins writing data to the new file.

Store operations, such as adding a session to the store (put store operations) or removing a session from the store (remove store operations), are queued in a buffer before they are written to the session store file. You can configure parameters that determine when the session store writes a queue to a session store file.

Session store files are deleted if they have not been modified and if no session activity has taken place for one week. All the data files that contain the sessions associated with a particular virtual router are deleted at the same time.

Active and Passive Session Stores

You can have a community of SAEs and duplicate session store data on each SAE in the community in case of an SAE failover. SAE communities are made up of SAEs that you configure as connected SAEs for a virtual router object.

SAEs in a community are given the role of either active SAE or passive SAE. The active SAE keeps session data up to date within the community. Each active session store opens a Transmission Control Protocol (TCP) connection to its passive SAE. The TCP connection triggers the creation of a passive session store in that SAE. When the active session store writes operations to the session store file, it passes them to passive session stores on all SAEs in the community.

When you modify a community, wait for passive session stores on the new community members to be updated before you shut down the currently active SAE. Otherwise, if you add a new member to a community, and then a failover from the current active SAE to the new member is triggered immediately, the new member's session store may not have received all data from the active SAE's session store.

Standby SAEs

You can configure standby SAEs for a configuration that include JUNOSe routers. In a community of SAEs, a standby SAE can provide redundancy for the active SAE. The active SAE connects to the standby SAE through a COPS-PR connection on port 3228. The active SAE maintains a separate session store connection with the standby SAE through port 8820 (default).

The active SAE replicates state and session data, including COPS messages received from JUNOSe routers, to the standby SAE. This replication reduces the failover time from one SAE to another. The active SAE detects a connection failure when a subsequent COPS message needs to be replicated because it has to wait for the standby to respond to the replication message. Both the active and standby SAEs detect a connection failure when the keep-alive timeout occurs (1 second).



NOTE: We recommend that you use a highly reliable and available connection between an active SAE and a standby SAE to ensure availability of the two SAEs.

For standby SAEs, you configure an SAE community and the session store at the same time by configuring SAE identifiers for in the configuration for the shared network device virtual router. In the configuration, an exclamation point identifies standby SAEs.

Session Store File Rotation

The session store periodically rotates the session store files. During rotation, the session store copies put store operations for live sessions from the oldest file to the end of the newest file. (Live sessions are sessions that have been created but not yet deleted.) It then deletes the oldest file. Sessions are rotated in batches, and you can configure the number of sessions that are rotated at the same time, and how much disk space is used by live sessions before files are rotated. No session store activity can take place while a batch of sessions is rotated.

- Related Topics**
- COPS Connection Between JUNOSe Routers and the SAE on page 45
 - Configuring the Session Store Feature (SRC CLI) on page 25
 - Adding JUNOSe Routers and Virtual Routers (SRC CLI) on page 46

- SRC Data Storage
- Configuring the Number of Threads for Sessions (SRC CLI) on page 29
- Viewing Statistics for Subscriber and Service Sessions (SRC CLI)
- Viewing Information About Subscriber Sessions by Session ID (C-Web Interface)

Configuring the Session Store Feature (SRC CLI)

You can configure three things for the session store feature:

1. Configuring Session Store Parameters for a Device Driver on page 25
2. Configuring Global Session Store Parameters on page 27
3. Reducing the Size of Objects for the Session Store Feature on page 28

Configuring Session Store Parameters for a Device Driver

Use the following configuration statements to configure session store parameters within a device driver configuration:

```
shared sae configuration driver ( aaa | junos | junose | pcmm | simulated | third-party )
  session-store {
    maximum-queue-age maximum-queue-age ;
    maximum-queued-operations maximum-queued-operations ;
    maximum-queue-size maximum-queue-size ;
    maximum-file-size maximum-file-size ;
    minimum-disk-space-usage minimum-disk-space-usage ;
    rotation-batch-size rotation-batch-size ;
    maximum-session-size maximum-session-size ;
    disk-load-buffer-size disk-load-buffer-size ;
    network-buffer-size network-buffer-size ;
    retry-interval retry-interval ;
    communications-timeout communications-timeout ;
    load-timeout load-timeout ;
    idle-timeout idle-timeout ;
    maximum-backlog-ratio maximum-backlog-ratio ;
    minimum-backlog minimum-backlog ;
  }
```

To configure session store parameters within a device driver configuration:

1. From configuration mode, access the configuration statement that configures the session store for your device driver. In this sample procedure, the session store for a JUNOS device driver is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration driver junos session-store
```

2. (Optional) Specify the maximum age that a queue of buffered store operations (such as adding a session to the store or removing a session from the store) can reach before the queue is written to a session store file.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-queue-age maximum-queue-age
```

3. (Optional) Specify the number of buffered store operations that are queued before the queue is written to a session store file.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-queued-operations maximum-queued-operations
```

4. (Optional) Specify the maximum size that a queue of buffered store operations can reach before the queue is written to a session store file.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-queue-size maximum-queue-size
```

5. (Optional) Specify the maximum size of session store files.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-file-size maximum-file-size
```

6. (Optional) Specify the percentage of space in all session store files that is used by live sessions.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set minimum-disk-space-usage minimum-disk-space-usage
```

7. (Optional) Specify the number of sessions that are rotated from the oldest file to the newest file at the same time that the oldest session store file is rotated.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set rotation-batch-size rotation-batch-size
```

8. (Optional) Specify the maximum size of a single subscriber or service session.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-session-size maximum-session-size
```

9. (Optional) Specify the size of the buffer that is used to load all of a session store's files from disk at startup.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set disk-load-buffer-size disk-load-buffer-size
```

10. (Optional) Specify the size of the buffer that holds messages or message segments that are waiting to be sent to passive session stores.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set network-buffer-size network-buffer-size
```

11. (Optional) Specify the time interval between attempts by the active session store to connect to missing passive session stores.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set retry-interval retry-interval
```

12. (Optional) Specify the amount of time that a session store waits before closing when it is blocked from reading or writing a message.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set communications-timeout communications-timeout
```

13. (Optional) Specify the time that an active session store waits for a passive session store or a passive session store waits for an active session store to load its data from disk before it closes the connection to the session store.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set load-timeout load-timeout
```

14. (Optional) Specify the time that a passive session store waits for activity from the active session store before it closes the connection to the active session store.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set idle-timeout idle-timeout
```

15. (Optional) Specify when the active session store closes the connection to a passive session store because of a backlog of messages waiting to be sent.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-backlog-ratio maximum-backlog-ratio
```

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set minimum-backlog minimum-backlog
```

16. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# show
maximum-queue-age 5000;
maximum-queued-operations 50;
maximum-queue-size 51050;
maximum-file-size 25000000;
minimum-disk-space-usage 25;
rotation-batch-size 50;
maximum-session-size 10000;
disk-load-buffer-size 1000000;
network-buffer-size 51050;
retry-interval 5000;
communications-timeout 60000;
load-timeout 420000;
idle-timeout 3600000;
maximum-backlog-ratio 1.5;
minimum-backlog 5000000;
```

Configuring Global Session Store Parameters

This topic describes how to configure global session store parameters that are shared by all session store instances (active or passive) on the SAE. You can also configure session store parameters within a device driver configuration. See “Configuring the Session Store Feature (SRC CLI)” on page 25.

Use the following configuration statements to configure global session store parameters.

```
shared sae configuration driver session-store {
```

```
ip-address ip-address ;  
port port ;  
root-directory root-directory ;  
}
```

To configure global session store parameters:

1. From configuration mode, access the configuration statement that configures the global session store parameters. In this sample procedure, the global session store is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration driver session-store
```

2. (Optional) Specify the IP address or hostname that the session store infrastructure on this SAE uses to listen for incoming TCP connections from active session stores.

```
[edit shared sae group se-region configuration driver session-store]  
user@host# set ip-address ip-address
```

3. (Optional) Specify the TCP port number on which the session store infrastructure on this SAE listens for incoming connections from active session stores.

```
[edit shared sae group se-region configuration driver session-store]  
user@host# set port port
```

4. (Optional) Specify the root directory in which the session store creates files.

```
[edit shared sae group se-region configuration driver session-store]  
user@host# set root-directory root-directory
```

5. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration driver session-store]  
user@host# show  
ip-address 10.10.70.0;  
port 8820;  
root-directory var/sessionStore;
```

Reducing the Size of Objects for the Session Store Feature

You can use serialized data compression to reduce the size of sessions objects that the SAE sends across the network for the session store feature. Enabling this property reduces the size of objects, but increases the CPU load on the SAE.

Use the following configuration statement to specify whether or not session objects are compressed.

```
shared sae configuration {  
  compress-session-data;  
}
```

To specify whether or not session objects are compressed:

1. From configuration mode, access the sae configuration. In this sample procedure, data compression is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration
```

2. Enable reducing the size of session objects (subscriber and service sessions) that the SAE sends across the network for the session store feature.

```
[edit shared sae group se-region configuration]
user@host# set compress-session-data
```

3. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration]
user@host# show compress-session-data
compress-session-data;
```

Configuring the Number of Threads for Sessions (SRC CLI)

Use the following configuration statement to set the number of threads used for session-related activity.

```
shared sae configuration session-job-manager {
  number-of-threads number-of-threads;
}
```

To configure the number of threads used to handle session-related activity:

1. From configuration mode, access the session job manager configuration. In this sample procedure, the number of threads is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration session-job-manager
```

2. Specify the number of threads used for session-related activity.

```
[edit shared sae group se-region configuration session-job-manager]
user@host# set number-of-threads number-of-threads
```

3. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration session-job-manager]
user@host# show
number-of-threads 10;
```

- Related Topics**
- Configuring the Session Store Feature (SRC CLI) on page 25
 - Storing Subscriber and Service Session Data on page 23

CHAPTER 4

Managing SAE Data (SRC CLI)

- Commands to Manage SAE Data on page 31
- Reloading the SAE Data (SRC CLI) on page 32
- Reloading the SAE Configuration (SRC CLI) on page 32
- Reloading Services (SRC CLI) on page 33
- Reloading Subscriptions (SRC CLI) on page 33
- Reloading Interface Classification Scripts (SRC CLI) on page 33
- Reloading Domain Maps (SRC CLI) on page 33
- Removing the Directory Blacklist (SRC CLI) on page 34
- Removing Login Registrations (SRC CLI) on page 34
- Removing Equipment Registrations (SRC CLI) on page 35
- Modifying Failover Server Parameters (SRC CLI) on page 35
- Shutting Down the Device Drivers (SRC CLI) on page 36

Commands to Manage SAE Data

You can use the following operational mode commands to manage SAE data:

- **clear sae directory-blacklist**
- **clear sae registered equipment**
- **clear sae registered login**
- **request sae load configuration**
- **request sae load domain-map**
- **request sae load interface-classification**
- **request sae load services**
- **request sae load subscriptions**
- **request sae modify device failover**
- **request sae shutdown device**
- **show sae directory-blacklist**

- **show sae drivers**
- **show sae registered equipment**
- **show sae registered login**

For detailed information about each command, see the *SRC CLI Command Reference*.

- Related Topics**
- Reloading the SAE Data (SRC CLI) on page 32
 - Reloading the SAE Configuration (SRC CLI) on page 32

Reloading the SAE Data (SRC CLI)

You can reload specified configuration components. You can reload the SAE server's current configuration for:

- SAE configuration
- Services
- Subscriptions
- Interface classifiers
 - Domain map

- Related Topics**
- Viewing Information About SAE Interfaces (SRC CLI)
 - Viewing Information About SAE Device Drivers (SRC CLI)
 - Viewing Information About Services (SRC CLI)
 - Viewing Information About Policies on the SAE (SRC CLI)

Reloading the SAE Configuration (SRC CLI)

To reload the SAE configuration data from the directory:

```
user@host> request sae load configuration
```

The new configuration takes effect immediately.

- Related Topics**
- Initially Configuring the SAE
 - Reloading the SAE Data (SRC CLI) on page 32
 - For more information about monitoring the SAE data with the SRC CLI, see Viewing Information About the Directory Blacklist (SRC CLI)
 - Reloading Services (SRC CLI) on page 33

Reloading Services (SRC CLI)

To reload the services, scopes, virtual routers, policies, service mutex groups, and service schedules from the directory:

```
user@host> request sae load services
```

Related service sessions are activated, deactivated, or reactivated as needed.

- Related Topics**
- Reloading the SAE Configuration (SRC CLI) on page 32
 - Viewing Information About Services (SRC CLI)
 - Viewing Information About Services (C-Web Interface)
 - Commands to Manage SAE Data on page 31
 - Reloading the SAE Data (SRC CLI) on page 32
 - Reloading Subscriptions (SRC CLI) on page 33

Reloading Subscriptions (SRC CLI)

To reload all subscriptions from the directory:

```
user@host> request sae load subscriptions
```

Related service sessions are activated, deactivated, or reactivated as needed.

- Related Topics**
- Reloading the SAE Configuration (SRC CLI) on page 32
 - Viewing Statistics for Subscriber and Service Sessions (SRC CLI)
 - For more information about viewing subscriber sessions with the SRC CLI, see Viewing General Information for Subscriber Sessions (SRC CLI)
 - Reloading the SAE Data (SRC CLI) on page 32

Reloading Interface Classification Scripts (SRC CLI)

To reload the interface classification scripts from the directory, and apply the result of the interface classification changes to the router:

```
user@host> request sae load interface-classification
```

- Related Topics**
- Viewing Information About SAE Interfaces (SRC CLI)
 - Reloading the SAE Data (SRC CLI) on page 32
 - Commands to Manage SAE Data on page 31

Reloading Domain Maps (SRC CLI)

To reload the mapping of domain names to retailer entries:

```
user@host> request sae load domain-map
```

This mapping is made available to the SAE's subscriber classification script.

- Related Topics**
- Reloading the SAE Data (C-Web Interface) on page 37
 - Reloading Subscriptions (SRC CLI) on page 33
 - Commands to Manage SAE Data on page 31

Removing the Directory Blacklist (SRC CLI)

To remove the directory blacklist:

1. Issue the **show sae directory-blacklist** command to view information about the directory blacklist.
2. Issue the **clear sae directory-blacklist command** to remove the directory blacklist.

- Related Topics**
- Removing the Directory Blacklist (C-Web Interface) on page 39
 - Removing Login Registrations (SRC CLI) on page 34
 - Viewing Information About the Directory Blacklist (SRC CLI)
 - Commands to Manage SAE Data on page 31
 - Reloading the SAE Data (SRC CLI) on page 32

Removing Login Registrations (SRC CLI)

You can delete all login registrations, or you can delete a specific registration.

To remove login registrations:

1. Issue the **show sae registered login** command to view the login registrations.
 2. Issue the **clear sae registered login command** to remove all login registrations.
- To remove a specific registration, use the **mac-address** option and specify the media access control (MAC) address for the registration.

```
user@host> clear sae registered login mac-address mac-address
```

- To specify that no confirmation is requested before the software deletes the registration entries, use the **force** option.

```
user@host> clear sae registered login force
```

```
user@host> clear sae registered login mac-address mac-address force
```

- Related Topics**
- Removing Login Registrations (C-Web Interface) on page 40
 - Removing Equipment Registrations (SRC CLI) on page 35
 - Viewing Login Registrations (SRC CLI)

- Viewing Login Registrations (C-Web Interface)
- Reloading the SAE Data (SRC CLI) on page 32

Removing Equipment Registrations (SRC CLI)

You can delete all equipment registrations, or you can delete a specific registration. The demonstration residential portal included with the SRC Application Library provides an example of how to use equipment registration.

To remove equipment registrations:

1. Issue the **show sae registered equipment** command to view the equipment registrations.
 2. Issue the **clear sae registered equipment** command to remove all equipment registrations.
- To remove a specific registration, use the **mac-address** option and specify the media access control (MAC) address for the registration.

```
user@host> clear sae registered equipment mac-address mac-address
```

- To specify that no confirmation is requested before the software deletes the registration entries, use the **force** option.

```
user@host> clear sae registered equipment force
user@host> clear sae registered equipment mac-address mac-address force
```

Related Topics

- Removing Equipment Registrations (C-Web Interface) on page 40
- Removing Login Registrations (SRC CLI) on page 34
- Viewing Equipment Registrations (SRC CLI)
- Viewing Equipment Registrations (C-Web Interface)
- Reloading the SAE Data (SRC CLI) on page 32

Modifying Failover Server Parameters (SRC CLI)

To modify failover server parameters:

1. Issue the **show sae drivers brief** command to view the router or device instances.
 2. Issue the **request sae modify device failover virtual-router-name virtual-router-name command** to modify failover server parameters.
- (Optional) To modify the IP address of an alternate SAE server to which a router can reconnect when this driver closes its connection, use the **ip-address** option. This option is not applicable to the PCMM device driver.

```
user@host> request sae modify device failover virtual-router-name
virtual-router-name ip-address ip-address
```

- (Optional) To modify the port of an alternate SAE server to which a router can reconnect when this driver closes its connection, use the **tcp-port** option. This option is not applicable to the PCMM device driver.

```
user@host> request sae modify device failover virtual-router-name  
virtual-router-name tcp-port tcp-port
```

- (Optional) To specify whether the device driver sends its own failover IP address and port to the router when it closes its connection, use the **use-failover-server** option. This option is not applicable to the PCMM device driver.

```
user@host> request sae modify device failover virtual-router-name  
virtual-router-name use-failover-server
```

- (Optional) To specify that no confirmation is requested before the software modifies the parameters, use the **force** option.

```
user@host> request sae modify device failover virtual-router-name  
virtual-router-name force  
user@host> request sae modify device failover virtual-router-name  
virtual-router-name ip-address ip-address force  
user@host> request sae modify device failover virtual-router-name  
virtual-router-name tcp-port tcp-port force  
user@host> request sae modify device failover virtual-router-name  
virtual-router-name use-failover-server force
```

- Related Topics**
- Modifying Failover Server Parameters (C-Web Interface) on page 41
 - Viewing Statistics for Device Drivers (SRC CLI)
 - Viewing Statistics for Specific Device Drivers (SRC CLI)
 - Viewing Information About Device Drivers (C-Web Interface)

Shutting Down the Device Drivers (SRC CLI)

To shut down the specified router or device instance:

1. Issue the **show sae drivers brief** command to view the router or device instances.
2. Issue the **request sae shutdown device command** to shut down all device drivers.
 - To shut down specific drivers managing a virtual router, use the **filter** option and specify all or part of the name of the virtual router.

```
user@host> request sae shutdown device filter filter
```

- To specify that no confirmation is requested before the software shuts down the device drivers, use the **force** option.

```
user@host> request sae shutdown device force  
user@host> request sae shutdown device filter filter force
```

- Related Topics**
- Shutting Down the Device Drivers (C-Web Interface) on page 41
 - Viewing Statistics for Device Drivers (SRC CLI)

CHAPTER 5

Managing SAE Data (C-Web Interface)

- Reloading the SAE Data (C-Web Interface) on page 37
- Reloading the SAE Configuration (C-Web Interface) on page 37
- Reloading Services (C-Web Interface) on page 38
- Reloading Subscriptions (C-Web Interface) on page 38
- Reloading Interface Classification Scripts (C-Web Interface) on page 39
- Reloading Domain Maps (C-Web Interface) on page 39
- Removing the Directory Blacklist (C-Web Interface) on page 39
- Removing Login Registrations (C-Web Interface) on page 40
- Removing Equipment Registrations (C-Web Interface) on page 40
- Modifying Failover Server Parameters (C-Web Interface) on page 41
- Shutting Down the Device Drivers (C-Web Interface) on page 41

Reloading the SAE Data (C-Web Interface)

You can reload specified configuration components. You can reload the SAE server's current configuration for:

- SAE configuration
- Services
- Subscriptions
- Interface classifiers
- Domain map

Reloading the SAE Configuration (C-Web Interface)

To reload the SAE configuration data from the directory:

1. Click **Manage>Request>SAE>Load>Configuration**.
The Configuration pane appears.
2. Enter information as described in the Help text in the main pane, and click **OK**.

The new configuration takes effect immediately.

- Related Topics**
- Initially Configuring the SAE (C-Web Interface)
 - Reloading the SAE Data (C-Web Interface) on page 37
 - For more information about monitoring the SAE data with the C-Web interface, see Viewing Information About the Directory Blacklist (C-Web Interface)
 - Reloading Services (C-Web Interface) on page 38

Reloading Services (C-Web Interface)

To reload the services, scopes, virtual routers, policies, service mutex groups, and service schedules from the directory:

1. Click **Manage>Request>SAE>Load>Services**.
The Services pane appears.
2. Enter information as described in the Help text in the main pane, and click **OK**.

Related service sessions are activated, deactivated, or reactivated as needed.

- Related Topics**
- Reloading the SAE Configuration (C-Web Interface) on page 37
 - Viewing Information About Services (C-Web Interface)
 - Commands to Manage SAE Data on page 31
 - Reloading the SAE Data (C-Web Interface) on page 37
 - Reloading Subscriptions (C-Web Interface) on page 38

Reloading Subscriptions (C-Web Interface)

To reload all subscriptions from the directory:

1. Click **Manage>Request>SAE>Load>Subscriptions**.
The Subscriptions pane appears.
2. Enter information as described in the Help text in the main pane, and click **OK**.

Related service sessions are activated, deactivated, or reactivated as needed.

- Related Topics**
- Reloading the SAE Configuration (C-Web Interface) on page 37
 - Viewing Statistics for Subscriber and Service Sessions (SRC CLI)
 - For more information about viewing subscriber sessions with the SRC CLI, see Viewing General Information for Subscriber Sessions (SRC CLI)
 - Reloading the SAE Data (C-Web Interface) on page 37

Reloading Interface Classification Scripts (C-Web Interface)

To reload the interface classification scripts from the directory, and apply the result of the interface classification changes to the router:

1. Click **Manage>Request>SAE>Load>Interface Classification**.
The Interface Classification pane appears.
2. Enter information as described in the Help text in the main pane, and click **OK**.

- Related Topics**
- Viewing Information About SAE Interfaces (SRC CLI)
 - Reloading the SAE Data (C-Web Interface) on page 37
 - Commands to Manage SAE Data on page 31

Reloading Domain Maps (C-Web Interface)

To reload the mapping of domain names to retailer entries:

1. Click **Manage>Request>SAE>Load>Domain Map**.
The Domain Map pane appears.
2. Enter information as described in the Help text in the main pane, and click **OK**.

This mapping is made available to the SAE's subscriber classification script.

- Related Topics**
- Reloading the SAE Data (SRC CLI) on page 32
 - Viewing Information About Interfaces (C-Web Interface)
 - Viewing Information About Device Drivers (C-Web Interface)
 - Viewing Information About Services (C-Web Interface)
 - Viewing Information About Policies (C-Web Interface)

Removing the Directory Blacklist (C-Web Interface)

To remove the directory blacklist:

1. To view information about the directory blacklist:
 - a. Click **Monitor>SAE>Directory Blacklist**.
The Directory Blacklist pane appears.
 - b. Enter information as described in the Help text in the main pane, and click **OK**.
2. To remove the directory blacklist:
 - a. Click **Manage>Clear>SAE>Directory Blacklist**.

The Directory Blacklist pane appears.

- b. Enter information as described in the Help text in the main pane, and click **OK**.

- Related Topics**
- Removing the Directory Blacklist (SRC CLI) on page 34
 - Removing Login Registrations (C-Web Interface) on page 40
 - Viewing Information About the Directory Blacklist (C-Web Interface)
 - Reloading the SAE Data (C-Web Interface) on page 37

Removing Login Registrations (C-Web Interface)

You can delete all login registrations, or you can delete a specific registration.

To remove login registrations:

1. Click **Monitor>SAE>Registered>Login**.
The Login pane appears.
2. Enter information as described in the Help text in the main pane, and click **OK**.

To remove login registrations:

1. Click **Manage>Clear>SAE>Registered>Login**.
The Login pane appears.
2. Enter information as described in the Help text in the main pane, and click **OK**.

- Related Topics**
- Removing Login Registrations (SRC CLI) on page 34
 - Removing the Directory Blacklist (C-Web Interface) on page 39
 - Viewing Login Registrations (C-Web Interface)
 - Reloading the SAE Data (C-Web Interface) on page 37

Removing Equipment Registrations (C-Web Interface)

You can delete all equipment registrations, or you can delete a specific registration. The demonstration residential portal included with the SRC Application Library provides an example of how to use equipment registration.

To remove equipment registrations:

1. Click **Monitor>SAE>Registered>Equipment**.
The Equipment pane appears.
2. Enter information as described in the Help text in the main pane, and click **OK**.

To remove login registrations:

1. Click **Manage>Clear>SAE>Registered>Equipment**.
The Equipment pane appears.
2. Enter information as described in the Help text in the main pane, and click **OK**.

- Related Topics**
- Removing Equipment Registrations (SRC CLI) on page 35
 - Removing Login Registrations (C-Web Interface) on page 40
 - Viewing Equipment Registrations (C-Web Interface)
 - Reloading the SAE Data (C-Web Interface) on page 37

Modifying Failover Server Parameters (C-Web Interface)

To modify failover server parameters:

1. To view the router or device instances:
 - a. Click **Monitor>SAE>Drivers**.
The Drivers pane appears.
 - b. Enter information as described in the Help text in the main pane, and click **OK**.
2. To modify failover server parameters:
 - a. Click **Manage>SAE>Request>Modify>Device>Failover**.
The Failover pane appears.
 - b. Enter information as described in the Help text in the main pane, and click **OK**.

- Related Topics**
- Modifying Failover Server Parameters (SRC CLI) on page 35
 - Viewing Information About Device Drivers (C-Web Interface)
 - Viewing Statistics for Device Drivers (SRC CLI)
 - Viewing Statistics for Specific Device Drivers (SRC CLI)

Shutting Down the Device Drivers (C-Web Interface)

To shut down the specified router or device instance:

1. To view the router or device instances:
 - a. Click **Monitor>SAE>Drivers**.
The Drivers pane appears.
 - b. Enter information as described in the Help text in the main pane, and click **OK**.
2. To shut down all device drivers:

- a. Click **Manage>SAE>Request>Shutdown>Device**.

The Device pane appears.

- b. Enter information as described in the Help text in the main pane, and click **OK**.

- Related Topics**
- Shutting Down the Device Drivers (SRC CLI) on page 36
 - Viewing Information About Device Drivers (C-Web Interface)

PART 2

Using Juniper Networks Routers in the SRC Network

- Using JUNOSe Routers in the SRC Network (SRC CLI) on page 45
- Using JUNOS Routing Platforms in the SRC Network (SRC CLI) on page 65

CHAPTER 6

Using JUNOS^e Routers in the SRC Network (SRC CLI)

- COPS Connection Between JUNOS^e Routers and the SAE on page 45
- Adding JUNOS^e Routers and Virtual Routers (SRC CLI) on page 46
- Configuring the SAE to Manage JUNOS^e Routers (SRC CLI) on page 50
- How SNMP Obtains Information from Routers for the SRC Software on page 52
- Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 53
- Specifying JUNOS^e Router Initialization Scripts on the SAE (SRC CLI) on page 55
- Accessing the Router CLI on page 57
- Starting the SRC Client on a JUNOS^e Router on page 57
- Stopping the SRC Client on a JUNOS^e Router on page 58
- Monitoring Interactions Between the SAE and the Router Running JUNOS^e Software on page 58
- Troubleshooting Problems with Managing JUNOS^e Routers on page 59
- Viewing the State of JUNOS^e Device Drivers (SRC CLI) on page 60
- Viewing Statistics for Specific JUNOS^e Device Drivers (SRC CLI) on page 61
- Viewing Statistics for All JUNOS^e Device Drivers (SRC CLI) on page 61
- Viewing the State of JUNOS^e Device Drivers (C-Web Interface) on page 62
- Viewing Statistics for All JUNOS^e Device Drivers (C-Web Interface) on page 62

COPS Connection Between JUNOS^e Routers and the SAE

Configuring the SRC client on a JUNOS^e router opens a Common Open Policy Service (COPS) protocol layer connection to the SAE. When the SRC client software establishes a TCP/IP connection to the SAE, the SAE starts to manage the JUNOS^e router. Subsequently, the SRC client sends configuration changes made on the JUNOS^e router to the SAE, and the SAE updates SRC configurations for services and policies accordingly.

The SAE supports two versions of COPS:

- COPS usage for policy provisioning (COPS-PR)

- COPS External Data Representation Standard (COPS-XDR)

The version of COPS that you use depends on the version of COPS that your JUNOSe router supports. When you set up your JUNOSe router to work with the SAE, you enable either COPS-PR mode or COPS-XDR mode.

Highly Available Connections to JUNOSe Routers

JUNOSe routers maintain state information, a feature that allows an active, managing SAE to reconnect to a JUNOSe router without performing a data resynchronization in the following instances:

- The network connection between the SAE and the JUNOSe router is disrupted, and the router reconnects to the SAE
- For JUNOSe routers with high availability configured, when the secondary SRP takes control from a failed SRP it can reconnect to the SAE

To maintain highly available connections to JUNOSe routers, configure an SAE community and the session store by configuring SAE identifiers for in the configuration for the shared network device virtual router. In the configuration, an exclamation point identifies standby SAEs.

- Related Topics**
- Storing Subscriber and Service Session Data on page 23
 - Adding JUNOSe Routers and Virtual Routers (SRC CLI) on page 46
 - Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 53
 - How SNMP Obtains Information from Routers for the SRC Software on page 52
 - Configuring the SAE to Manage JUNOSe Routers (SRC CLI) on page 50
 - Starting the SRC Client on a JUNOSe Router on page 57
 - Monitoring Interactions Between the SAE and the Router Running JUNOSe Software on page 58

Adding JUNOSe Routers and Virtual Routers (SRC CLI)

The SAE uses router and virtual router objects to manage interfaces on JUNOSe virtual routers. Each JUNOSe router in the SRC network and its virtual routers (VRs) must have a configuration.

There are two ways to add routers:

1. Adding Operative JUNOSe Routers and Virtual Routers on page 47
2. Adding Routers Individually (SRC CLI) on page 47
3. Adding Virtual Routers Individually (SRC CLI) on page 48

Adding Operative JUNOSe Routers and Virtual Routers

To add routers and JUNOSe VRs that are currently operative and have an operating SNMP agent:

- In operational mode, enter the following command:

```
user@host> request network discovery network network <community community>
```

where:

- **network** —Address (with or without mask) of the network to discover
- **community** —Name of the SNMP community to which the devices belong

If you add a router using the discover network feature, the software adds the IP address of the first SNMP agent on the router to respond to the discover request.

After you add routers and JUNOSe VRs through network discovery, configure the virtual router's managing SAE address.

Adding Routers Individually (SRC CLI)

Use the following configuration statements to add a router:

```
shared network device name {
  description description ;
  management-address management-address ;
  device-type (junose| junos| pcmm| third-party);
  qos-profile [ qos-profile ...];
}
```

To add a router:

1. From configuration mode, access the configuration statements that configure network devices. You must specify the name of a device with lowercase characters. This procedure uses `junose_boston` as the name of the router.

```
user@host# edit shared network device junose_boston
```

The same procedure can be used for JUNOS routers.

2. (Optional) Add a description for the router.

```
[edit shared network device junose_boston]
user@host# set description description
```

3. (Optional) Add the IP address of the router.

```
[edit shared network device junose_boston]
user@host# set management-address management-address
```

4. (Optional) Specify the type of device that you are adding.

```
[edit shared network device junose_boston]
user@host# set device-type junose
```

5. (Optional) Specify quality of service (QoS) profiles that are configured on the router.

```
[edit shared network device junose_boston]
user@host# set qos-profile [ qos-profile ...]
```

6. (Optional) Verify your configuration.

```
[edit shared network device junose_boston]
user@host# show
description "Juniper Networks E320";
management-address 10.10.8.27;
device-type junose;
qos-profile dhcp-default;
interface-classifier {
  rule rule-0 {
    script #;
  }
}
```

Adding Virtual Routers Individually (SRC CLI)

Use the following configuration statements to add a virtual router:

```
shared network device name virtual-router name {
  sae-connection [ sae-connection ...];
  snmp-read-community snmp-read-community;
  snmp-write-community snmp-write-community;
  scope [ scope ...];
  local-address-pools local-address-pools;
  static-address-pools static-address-pools;
  tracking-plug-in [ tracking-plug-in ...];
  authentication-plug-in [ authentication-plug-in ...];
  vpn-id vpn-id;
}
```

To add a virtual router:

1. From configuration mode, access the configuration statements for virtual routers. You must specify the name of a device with lowercase characters. This procedure uses `junose_Boston` as the name of the router and `vr1` as the name of the virtual router.

```
user@host# edit shared network device junose_boston virtual-router vr1
```

2. Specify the addresses of SAEs that can manage this router. This step is required for the SAE to work with the router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set sae-connection [ sae-connection ...]
```

To specify the active SAE and the redundant SAE, enter an exclamation point (!) after the hostname or IP address of the connected SAE. For example:

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set sae-connection [sae1! sae2!]
```


3. (Optional) Specify an SNMP community name for SNMP read-only operations for this VR.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set snmp-read-community snmp-read-community
```

4. (Optional) Specify an SNMP community name for SNMP write operations for this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set snmp-write-community snmp-write-community
```

5. (Optional) Specify service scopes assigned to this virtual router. The scopes are available for subscribers connected to this virtual router for selecting customized versions of services.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set scope [scope ...]
```

6. (Optional) Specify the list of IP address pools that a virtual router currently manages and stores.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set local-address-pools local-address-pools
```

7. (Optional) Specify the list of IP address pools that a VR manages but does not store.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set static-address-pools static-address-pools
```

8. (Optional) Specify the plug-ins that track interfaces that the SAE manages on this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set tracking-plugin-in [tracking-plugin-in ...]
user@host# set authentication-plugin-in [authentication-plugin-in ...]
```

9. (Optional) Specify the VPN identifier used by this virtual router. For edge devices, you can specify VRF instead of a string to use the VRF instance reported by the device as the VPN identifier. For example, if you specify VRF for a JUNOSe router, the VPN identifier is the name of the virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set vpn-id (vpn-id | VRF)
```

10. (Optional) Verify your configuration.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# show
sae-connection 192.168.10.25;
  snmp-read-community *****;
  snmp-write-community *****;
  scope POP-Boston;
  local-address-pools "(10.25.8.0 10.25.20.255)";
```

```
static-address-pools "({10.30.30.0/24,10.30.30.0,10.30.30.255})";  
tracking-plugin flexRadius;
```

- Related Topics**
- Configuring the SAE to Manage JUNOS Routers (SRC CLI) on page 50
 - Specifying JUNOS Router Initialization Scripts on the SAE (SRC CLI) on page 55
 - Configuring Service Scopes (SRC CLI)
 - Types of Tracking Plug-Ins
 - Overview of Classification Scripts

Configuring the SAE to Manage JUNOS Routers (SRC CLI)

To set up the SAE to manage JUNOS routers, configure a router driver that specifies a COPS server that can accept COPS connections from the COPS client in JUNOS routers.

Use the following configuration statements to configure the SAE to manage JUNOS routers:

```
shared sae configuration driver junos {  
  cops-server-port cops-server-port ;  
  backlog backlog ;  
  keepalive-interval keepalive-interval ;  
  message-timeout message-timeout ;  
  cops-message-maximum-length cops-message-maximum-length ;  
  cops-message-read-buffer-size cops-message-read-buffer-size ;  
  cops-message-write-buffer-size cops-message-write-buffer-size ;  
  pending-address-timeout pending-address-timeout ;  
  cops-handler-threads cops-handler-threads ;  
  cached-driver-expiration cached-driver-expiration ;  
  drop-unmanaged-interfaces-xdr-driver;  
  track-unmanaged-interfaces-xdr-driver;  
}
```

To configure the SAE to manage JUNOS routers:

1. From configuration mode, access the configuration statement that configures the JUNOS router driver. In this sample procedure, the JUNOS driver is configured in the west-region group.

```
user@host# edit shared sae group west-region configuration driver junos
```

2. Configure the port number of the SAE COPS server. The port number must match the configuration of the SRC client in the JUNOS router.

```
[edit shared sae group west-region configuration driver junos]  
user@host# set cops-server-port cops-server-port
```

3. Configure the number of outstanding connection attempts before connections are dropped.

```
[edit shared sae group west-region configuration driver junos]  
user@host# set backlog backlog
```

4. Configure the interval between keepalive messages sent from the COPS client (the JUNOSe router).

```
[edit shared sae group west-region configuration driver junose]
user@host# set keepalive-interval keepalive-interval
```

5. Configure the timeout interval in which the COPS server waits for a response to COPS requests.

```
[edit shared sae group west-region configuration driver junose]
user@host# set message-timeout message-timeout
```

6. Configure the maximum length of a COPS message.

```
[edit shared sae group west-region configuration driver junose]
user@host# set cops-message-maximum-length cops-message-maximum-length
```

7. Configure the buffer size for receiving COPS messages from the JUNOSe client. We recommend that you use the default setting unless you are instructed to change it by Juniper Networks.

```
[edit shared sae group west-region configuration driver junose]
user@host# set cops-message-read-buffer-size cops-message-read-buffer-size
```

8. Configure the buffer size for sending COPS messages to the JUNOSe client. We recommend that you use the default setting unless you are instructed to change it by Juniper Networks.

```
[edit shared sae group west-region configuration driver junose]
user@host# set cops-message-write-buffer-size cops-message-read-buffer-size
```

9. Configure the maximum time that a DHCP address request remains pending.

```
[edit shared sae group west-region configuration driver junose]
user@host# set pending-address-timeout pending-address-timeout
```

10. Configure the size of the thread pool for handling unsolicited messages. These threads are shared among all JUNOSe router drivers.

```
[edit shared sae group west-region configuration driver junose]
user@host# set cops-handler-threads cops-handler-threads
```

11. Configure the minimum amount of time to keep the state of a router driver after its COPS connection has been closed.

```
[edit shared sae group west-region configuration driver junose]
user@host# set cached-driver-expiration cached-driver-expiration
```

12. (Optional) If you are using COPS-XDR, specify whether or not the JUNOSe router driver keeps a record of unmanaged interfaces.

```
[edit shared sae group west-region configuration driver junose]
user@host# set drop-unmanaged-interfaces-xdr-driver
```

13. (Optional) Enable or disable sending of interface-tracking events for unmanaged interfaces for the XDR router driver.

```
[edit shared sae group west-region configuration driver junose]
user@host# set track-unmanaged-interfaces-xdr-driver
```

14. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration driver junose]
user@host# show
cops-server-port 3288;
backlog 50;
keepalive-interval 45;
message-timeout 120000;
cops-message-maximum-length 200000;
cops-message-read-buffer-size 30000;
cops-message-write-buffer-size 30000;
pending-address-timeout 5000;
cops-handler-threads 20;
cached-driver-expiration 600;
drop-unmanaged-interfaces-xdr-driver;
track-unmanaged-interfaces-xdr-driver;
```

- Related Topics**
- [Creating Grouped Configurations for the SAE \(SRC CLI\)](#)
 - [Configuring the SAE to Manage JUNOSe Routers \(C-Web Interface\)](#)
 - [Monitoring Interactions Between the SAE and the Router Running JUNOSe Software on page 58](#)
 - [Troubleshooting Problems with Managing JUNOSe Routers on page 59](#)
 - [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 53](#)

How SNMP Obtains Information from Routers for the SRC Software

Some scripts in the SRC software use SNMP to get information from the router. For example, the **poolPublisher** router initialization script uses SNMP to read the IP pools.

- On the router, you can configure access to the router's SNMP server. See [Configuring the SNMP Server on the JUNOSe Router](#).
- On the SAE, you can configure global default SNMP communities that are used for read and write access to the router.
- You can specify SNMP communities for each virtual router. We recommend that you specify communities for each virtual router instead of configuring global communities.

- Related Topics**
- [Accessing the Router CLI on page 57](#)
 - [Configuring the SNMP Server on the JUNOSe Router](#)
 - [Configuring Global SNMP Communities in the SRC Software \(SRC CLI\) on page 107](#)
 - [Configuring Global SNMP Communities in the SRC Software \(C-Web Interface\)](#)

- Adding JUNOS Routers and Virtual Routers (SRC CLI) on page 46
- Adding JUNOS Routers and Virtual Routers (C-Web Interface)
- Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 53
- Specifying JUNOS Router Initialization Scripts on the SAE (SRC CLI) on page 55

Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers

When the SAE establishes a connection with a router or network device, it can run an initialization script to customize the setup of the connection. These initialization scripts are run when the connection between a router or network device and the SAE is established and again when the connection is dropped.

We provide the `lorPublisher` script in the `/opt/UMC/sae/lib` folder. The `lorPublisher` script publishes the interoperable object reference (IOR) of the SAE in the directory so that a NIC can associate a router with an SAE.

For JUNOS VRs that supply IP addresses from a local pool, a router initialization script is provided that identifies which VR supplies each IP pool and writes the information to the configuration. The SAE runs the script only when a COPS connection is established to the JUNOS router. Consequently, if you modify information about IP pools on a VR after the COPS connection is established, the SAE will not automatically register the changes, and you must update the configuration.

Table 3 on page 53 describes the router initialization scripts that we provide with the SRC software in the `/opt/UMC/sae/lib` folder.

Table 3: Router Initialization Scripts

Script Name	Function	When to Use Script
<code>lorPublisher</code>	Publishes the IOR of the SAE into an internal part of the shared configuration so that a NIC can associate a router with an SAE.	Use with JUNOS routers that do not supply IP addresses from local pools, and with JUNOS routing platforms. Use with all JUNOS routing platforms. Use with third-party network devices.
<code>poolPublisher</code>	Publishes the IOR of the SAE and local IP address pools in the directory so that a NIC can associate a router with an SAE and resolve the IP-to-SAE mapping.	Use with JUNOS virtual routers that supply IP addresses from local pools.

Interface Object Fields

Router initialization scripts are written in the Python programming language (www.python.org) and executed in the Jython environment (www.jython.org).

Router initialization scripts interact with the SAE through an interface object called `Ssp`. The SAE exports a number of fields through the interface object to the script and expects the script to provide the entry point to the SAE.

Table 4 on page 54 describes the fields that the SAE exports.

Table 4: Exported Fields

Ssp Attribute	Description
<code>Ssp.properties</code>	System properties object (class: <code>java.util.Properties</code>)—The properties should be treated as read-only by the script.
<code>Ssp.errorLog</code>	Error logger—Use the <code>Ssp.errorLog.println (message)</code> to send error messages to the log.
<code>Ssp.infoLog</code>	Info logger—Use the <code>Ssp.infoLog.println (message)</code> to send informational messages to the log.
<code>Ssp.debugLog</code>	Debug logger—Use the <code>Ssp.debugLog.println (message)</code> to send debug messages to the log.

The router initialization script must set the field `Ssp.routerInit` to a factory function that instantiates a router initialization object:

- `<VRName>`—Name of the virtual router in which the COPS client has been configured, format: `virtualRouterName@RouterName`
- `<virtualIp>`—Virtual IP address of the SAE (string, dotted decimal; for example: 192.168.254.1)
- `<realIp>`—Real IP address of the SAE (string, dotted decimal; for example, 192.168.1.20)
- `<VRip>`—IP address of the virtual router (string, dotted decimal)
- `<transportVR>`—Name of the virtual router used for routing the COPS connection, or `None`, if the COPS client is directly connected

The factory function must implement the following interface:

```
Ssp.routerInit(VRName,
virtualIp,
realIp,
VRip,
transportVR)
```

The factory function returns an interface object that is used to set up and tear down a connection for a given COPS server. A common case of a factory function is the constructor of a class.

The factory function is called directly after a COPS server connection is established. In case of problems, an exception should be raised that leads to the termination of the COPS connection.

Required Methods

Instances of the interface object must implement the following methods:

- *setup()*—Is called when the COPS server connection is established and is operational. In case of problems, an exception should be raised that leads to the termination of the COPS connection.
- *shutdown()*—Is called when the COPS server connection to the virtual router is terminated. This method should not raise any exceptions in case of problems.

Example: Router Initialization Script

The following script defines a router initialization class named *SillyRouterInit*. The interface class does not implement any useful functionality. The interface class just writes messages to the infoLog when the router connection is created or terminated.

```
class SillyRouterInit:
    def __init__(self, vrName, virtualIp, realIp, vrIp, transportVr):
        """ initialize router initialization object """
        self.vrName = vrName
        Ssp.infoLog.println("SillyRouterInit created")
    def setup(self):
        """ initialize connection to router """
        Ssp.infoLog.println("Setup connection to VR %(vrName)s" %
                           vars(self))
    def shutdown(self):
        """ shutdown connection to router """
        Ssp.infoLog.println("Shutdown connection to VR %(vrName)s" %
                           vars(self))
#
# publish interface object to Ssp core
#
Ssp.routerInit = SillyRouterInit
```

- Related Topics**
- How SNMP Obtains Information from Routers for the SRC Software on page 52
 - Specifying JUNOSe Router Initialization Scripts on the SAE (SRC CLI) on page 55
 - Accessing the Router CLI on page 57
 - Viewing Statistics for Specific JUNOSe Device Drivers (SRC CLI) on page 61
 - Troubleshooting Problems with Managing JUNOSe Routers on page 59

Specifying JUNOSe Router Initialization Scripts on the SAE (SRC CLI)

Use the following configuration statements to specify router initialization scripts for JUNOSe routers:

```
shared sae configuration driver scripts {
    extension-path extension-path ;
    general general ;
    junose-pr junose-pr ;
    junose-xdr junose-xdr ;
}
```

To configure router initialization scripts for JUNOSe routers:

1. From configuration mode, access the configuration statements that configure router initialization scripts. In this sample procedure, the scripts are configured in the west-region group.

```
user@host# edit shared sae group west-region configuration driver scripts
```

2. Specify the script for JUNOSe routers when the JUNOSe driver uses COPS-PR mode when connecting to the SAE.

```
[edit shared sae group west-region configuration driver scripts]  
user@host# set junose-pr junose-pr
```

3. Specify the script for JUNOSe routers when the JUNOSe driver uses COPS-XDR mode when connecting to the SAE.

```
[edit shared sae group west-region configuration driver scripts]  
user@host# set junose-xdr junose-xdr
```

In COPS-XDR mode, the router does not send the network access server (NAS) IP address to the SAE. If your configuration requires this value, add the following line to a JUNOSe script:

```
import ERXnasip
```

4. Configure a router initialization script that can be used for all types of routers that the SRC module supports.

```
[edit shared sae group west-region configuration driver scripts]  
user@host# set general general
```

5. Configure a path to router initialization scripts that are not in the default location, `/opt/UMC/sae/lib`.

```
[edit shared sae group west-region configuration driver scripts]  
user@host# set extension-path extension-path
```

6. (Optional) Verify your router initialization script configuration.

```
[edit shared sae group west-region configuration driver scripts]  
user@host# show  
junose-xdr poolPublisher;
```

Related Topics

- Accessing the Router CLI on page 57
- Configuring the SAE to Manage JUNOSe Routers (SRC CLI) on page 50
- Monitoring Interactions Between the SAE and the Router Running JUNOSe Software on page 58
- Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 53

Accessing the Router CLI

You can access the CLIs of Juniper Networks routers through a Telnet or secure shell connection.

- To open a Telnet session to a router, use the **telnet** operational mode command. For example:

```
user@host> telnet 10.10.10.3
```

- To open a secure shell connection, use the **ssh** operational command. For example:

```
user@host> ssh host 10.10.10.3
```

Related Topics

- Specifying JUNOSe Router Initialization Scripts on the SAE (SRC CLI) on page 55
- Starting the SRC Client on a JUNOSe Router on page 57
- Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 53

Starting the SRC Client on a JUNOSe Router

JUNOSe routers use an SRC client to interact with the SAE. See *JUNOSe Broadband Access Configuration Guide* for complete information about configuring the SRC client on a JUNOSe router.

To start the SRC client:

1. Access the router CLI.
2. Access Global configuration mode.

```
host1# configure terminal
```

3. Switch to the virtual router for which you want to create an SRC client.

```
host1(config)#virtual-router <vrName>
```

4. Enable the SRC client.

To enable COPS-PR mode:

```
host1:<vrName>(config)#sscc enable cops-pr
```

To enable COPS-XDR mode:

```
host1:<vrName>(config)#sscc enable
```

5. Set the primary address from the configuration directory.

```
host1:<vrName>(config)#sscc primary address <ipAddress> port 3288
```

- Related Topics**
- Stopping the SRC Client on a JUNOSe Router on page 58
 - Accessing the Router CLI on page 57
 - Specifying JUNOSe Router Initialization Scripts on the SAE (SRC CLI) on page 55
 - Viewing Statistics for All JUNOSe Device Drivers (SRC CLI) on page 61

Stopping the SRC Client on a JUNOSe Router

JUNOSe routers use an SRC client to interact with the SAE. See *JUNOSe Broadband Access Configuration Guide* for complete information about configuring the SRC client on the JUNOSe router.

To stop the SRC client:

1. Access the router CLI.
See “Accessing the Router CLI” on page 57.
2. Access Global configuration mode.
`host1#configure terminal`
3. Switch to the virtual router for which you want to stop an SRC client.
`host1(config)#virtual-router <vrName>`
4. Disable the SRC client.
`host1:<vrName>(config)#no ssc enable`

- Related Topics**
- Starting the SRC Client on a JUNOSe Router on page 57
 - Viewing Statistics for All JUNOSe Device Drivers (SRC CLI) on page 61

Monitoring Interactions Between the SAE and the Router Running JUNOSe Software

Purpose Monitor connection between the SAE and a JUNOSe router.

Action To monitor the connection between the router and the SAE:

- Use the **show ssc info** command on the JUNOSe router

To display the version number of the SRC client:

- Use the **show ssc version** command on the JUNOSe router.

See the *JUNOSe Command Reference Guide* for details about these commands.

You can also monitor the interactions between the SRC module and the router in the log files for the SAE and in the log files generated by the JUNOSe router.

- For information about configuring logging on JUNOSe routers, see *JUNOSe System Event Logging Reference Guide*.

- Related Topics**
- Specifying JUNOSe Router Initialization Scripts on the SAE (SRC CLI) on page 55
 - Configuring the SAE to Manage JUNOSe Routers (SRC CLI) on page 50
 - Troubleshooting Problems with Managing JUNOSe Routers on page 59

Troubleshooting Problems with Managing JUNOSe Routers

Problem SRC client or JUNOSe router is not working as expected.

Solution You can troubleshoot problems with the SRC client on JUNOSe routers and with managed JUNOSe routers, interfaces, and services on the SAE.

To troubleshoot SRC problems on the router:

1. Look at the log files for the SAE and the log files generated by the SRC client on the JUNOSe router.
 - If the log files indicate a problem with specific interfaces on the router, review the configuration of the associated policies in the SRC module, and fix any errors.
 - If the log files indicate a problem with a specific service or its associated policy rules, review the configuration of the service or policies in the SRC module, and fix any errors.
 - If the log files indicate only that the SRC client is not responding, ensure that the values in the SAE configuration match the values in the SRC client configuration on the router.
2. Restart the SRC client on the JUNOSe router.

When you restart the SRC client, the SRC client removes all policies that were installed by the SRC module and reports all interfaces again.



NOTE: DHCP addresses that were managed are not reported again, so we recommend that you do not restart the SRC client if you are managing DHCP sessions.

To restart the SRC client in COPS-PR mode, enter the following commands:

```
host1:<vrName>(config)#no ssc enable
host1:<vrName>(config)#sscc enable cops-pr
```

To restart the SRC client in COPS-XDR mode, enter the following commands:

```
host1:<vrName>(config)#no ssc enable
host1:<vrName>(config)#sscc enable
```

If restarting the SRC client does not resolve the problem, rebuild the router configuration and restart the client.

- Related Topics**
- Monitoring Interactions Between the SAE and the Router Running JUNOS Software on page 58
 - Viewing the State of JUNOS Device Drivers (SRC CLI) on page 60
 - Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI) on page 61
 - Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 53

Viewing the State of JUNOS Device Drivers (SRC CLI)

Purpose Display the state of JUNOS drivers.

Action Use the following operational mode command:

```
show sae drivers <device-name device-name> < (brief) > <maximum-results  
maximum-results>
```

For example:

```
user@host> show sae drivers device-name default@dryad
JUNOS Driver
Device name                default@dryad
Device type                junose
Device IP                  10.227.7.244
Local IP                   10.227.7.172
TransportRouter            default@dryad
Device version             7.2.0
Start time                 Tue Feb 13 14:18:44 EST 2007
Number of notifications    20
Number of processed added  14
Number of processed changed 0
Number of processed deleted 6
Number of provisioning attempt 30
Number of provisioning attempt failed 0
Number of outstanding decisions 0
Number of SAP              7
Number of PAP              1
  Job Queue
  Size                     0
  Age (ms)                 1
  Total enqueued           28
  Total dequeued           28
  Average job time (ms) 426
  State Synchronization
  Number recovered subscriber sessions 0
  Number recovered service sessions    0
  Number recovered interface sessions  0
  Number invalid subscriber sessions    0
  Number invalid service sessions       0
  Number invalid interface sessions     0
  Background restoration start time     Tue Feb 13 14:18:49 EST 2007
  Background restoration end time       Tue Feb 13 14:18:49 EST 2007
```

```

Number subscriber sessions restored in background 0
Number of provisioning objects left to collect    0
Total number of provisioning objects to collect  11
Start time                                     Tue Feb 13 14:18:45 EST 2007

End time                                       Tue Feb 13 14:18:47 EST 2007

Number of synched contexts                    7
Number of post-sync jobs                      6

```

Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI)

Purpose Display statistics for a specific JUNOS device driver.

Action Use the following operational mode command:

```
show sae statistics device <name name> < (brief) >
```

For example:

```

user@host> show sae statistics device name default@dryad
SNMP Statistics
Add notification handle time      6
Change notification handle time   0
Client ID                        default@dryad
Delete notification handle time   0
Failover IP                      0.0.0.0
Failover port                    0
Handle message time              60
Job queue age                    0
Job queue time                   4
Number message send              158
Number of added jobs             9
Number of add notifications      4
Number of change notifications   0
Number of delete notifications   0
Number of managed interfaces     4
Number of message errors         0
Number of message timeouts       0
Number of removed jobs           9
Number of user session established 0
Number of user session removed   0
Router type                      JUNOS COPS
Up time                          172286
Using failover server            false

```

- Related Topics**
- Troubleshooting Problems with Managing JUNOS Routers on page 59
 - Viewing the State of JUNOS Device Drivers (SRC CLI) on page 60
 - Viewing Statistics for All JUNOS Device Drivers (SRC CLI) on page 61
 - Monitoring Interactions Between the SAE and the Router Running JUNOS Software on page 58

Viewing Statistics for All JUNOS Device Drivers (SRC CLI)

Purpose Display SNMP statistics for all JUNOS device drivers.

Action Use the following operational mode command:

```
show sae statistics device common junose-cops
```

For example:

```
user@host> show sae statistics device common junose-cops
SNMP Statistics
Driver type                JUNOSE COPS
Number of close requests   0
Number of connections accepted 2
Number of current connections 1
Number of open requests    2
Server address             0:0:0:0:0:0:0
Server port                3288
Time since last redirect   186703
```

- Related Topics**
- Troubleshooting Problems with Managing JUNOSe Routers on page 59
 - Viewing the State of JUNOSe Device Drivers (SRC CLI) on page 60
 - Viewing Statistics for Specific JUNOSe Device Drivers (SRC CLI) on page 61
 - Viewing Statistics for All JUNOSe Device Drivers (C-Web Interface) on page 62

Viewing the State of JUNOSe Device Drivers (C-Web Interface)

Purpose If the log files indicate a problem with a specific driver, review the configuration of the associated JUNOSe device driver with the C-Web interface.

- Action**
1. Click **Monitor>SAE>Drivers**.
The Drivers pane appears.
 2. Enter information as described in the Help text in the main pane, and click **OK**.
The Drivers pane displays information about the JUNOSe device driver.

- Related Topics**
- Troubleshooting Problems with Managing JUNOSe Routers on page 59
 - Monitoring Interactions Between the SAE and the Router Running JUNOSe Software on page 58
 - Viewing Statistics for All JUNOSe Device Drivers (C-Web Interface) on page 62

Viewing Statistics for All JUNOSe Device Drivers (C-Web Interface)

Purpose To view SNMP statistics for all JUNOSe device driver:

- Action**
1. Click **Monitor>SAE>Statistics>Device>Common**.
The Common pane appears.
 2. Enter information as described in the Help text in the main pane, and click **OK**.
The Common pane displays statistics for the JUNOSe device driver.

- Related Topics**
- [Troubleshooting Problems with Managing JUNOSe Routers on page 59](#)
 - [Viewing Statistics for Specific JUNOSe Device Drivers \(SRC CLI\) on page 61](#)
 - [Viewing the State of JUNOSe Device Drivers \(C-Web Interface\) on page 62](#)
 - [Viewing Statistics for All JUNOSe Device Drivers \(SRC CLI\) on page 61](#)

CHAPTER 7

Using JUNOS Routing Platforms in the SRC Network (SRC CLI)

- BEEP Connection Between JUNOS Routing Platforms and the SAE on page 65
- Adding JUNOS Routing Platforms and Virtual Routers (SRC CLI) on page 66
- Configuring the SAE to Manage JUNOS Routing Platforms (SRC CLI) on page 69
- Configuring Secure Connections Between the SAE and JUNOS Routing Platforms on page 72
- Adding the Server Certificate on the Routing Platform on page 72
- Creating a Client Certificate for the Router on page 73
- Adding the Client Certificate on the Router on page 73
- Configuring the SAE to Use TLS (SRC CLI) on page 74
- Configuring TLS on the SAE (SRC CLI) on page 74
- SAE Verification of JUNOS Configuration Changes on page 75
- Setting Up Periodic Configuration Checking (SRC CLI) on page 76
- Using SNMP to Retrieve Information from JUNOS Routers and JUNOS Routing Platforms (SRC CLI) on page 76
- Specifying Router Initialization Scripts on the SAE (SRC CLI) on page 77
- Configuring JUNOS Routing Platforms to Interact with the SAE on page 78
- SAE Tracking for LSPs Configured on JUNOS Routing Platforms on page 79
- Configuring the JUNOS Routing Platform to Apply Changes It Receives from the SAE on page 80
- Disabling Interactions Between the SAE and JUNOS Routing Platforms on page 81
- Monitoring Interactions Between the SAE and JUNOS Routing Platforms on page 81
- Troubleshooting Problems Between the SRC module and JUNOS Device Drivers on page 82

BEEP Connection Between JUNOS Routing Platforms and the SAE

For information about which JUNOS routing platforms and releases a particular SRC release supports, see the *SRC Release Notes*.

The SAE interacts with a JUNOS software process, referred to as the SRC software process in this documentation, on the JUNOS routing platform. The SAE and the SRC software process communicate using the Blocks Extensible Exchange Protocol (BEEP). You can secure the BEEP connection by using Transport Layer Security (TLS).

When the SRC software process establishes a BEEP session for the SAE, the SAE configures an interface on the JUNOS routing platform. The SAE builds the configuration for an interface using the policies stored in the directory. If the policies are subsequently modified, the SAE builds a new configuration and reconfigures the interface on the JUNOS routing platform. The JUNOS routing platform stores data about interfaces and services that the SAE manages in a configuration group called `sdx`. You must create this configuration group on the JUNOS routing platform.

- Related Topics**
- Adding Operative JUNOS Routing Platforms (C-Web Interface)
 - Configuring the SAE to Manage JUNOS Routing Platforms (SRC CLI) on page 69
 - Configuring Secure Connections Between the SAE and JUNOS Routing Platforms on page 72

Adding JUNOS Routing Platforms and Virtual Routers (SRC CLI)

On JUNOS routing platforms, the SAE manages interfaces. The SRC module associates a virtual router called `default` with each JUNOS routing platform. Each JUNOS routing platform in the SRC network and its associated virtual router (VR) called `default` must appear in the directory. The VRs are not actually configured on the JUNOS routing platform; the VR in the directory provides a way for the SAE to manage the interfaces on the JUNOS routing platform.

You can add routers the following ways:

- Adding Operative JUNOS Routing Platforms (SRC CLI) on page 66
- Adding Routers Individually (SRC CLI) on page 67
- Adding Virtual Routers Individually (SRC CLI) on page 68

Adding Operative JUNOS Routing Platforms (SRC CLI)

To add to the directory routers and JUNOS VRs that are currently operative and have an operating SNMP agent:

- In operational mode, enter the following command:

```
request network discovery network network <community community >
```

where:

- ***network*** —Address (with or without mask) of the network to discover
- ***community*** —Name of the SNMP community to which the devices belong

If you add a router using the `discover network` feature, the software adds the IP address of the first SNMP agent on the router to respond to the `discover` request.

Adding Routers Individually (SRC CLI)

Use the following configuration statements to add a router:

```
shared network device name {
  description description ;
  management-address management-address ;
  device-type (junose| junos| pcmm| third-party);
  qos-profile [ qos-profile ...];
}
```

To add a router:

1. From configuration mode, access the configuration statements that configure network devices. You must specify the name of a device with lowercase characters. This procedure uses `junose_boston` as the name of the router.

```
user@host# edit shared network device junose_boston
```

The same procedure can be used for JUNOS routers.

2. (Optional) Add a description for the router.

```
[edit shared network device junose_boston]
user@host# set description description
```

3. (Optional) Add the IP address of the router.

```
[edit shared network device junose_boston]
user@host# set management-address management-address
```

4. (Optional) Specify the type of device that you are adding.

```
[edit shared network device junose_boston]
user@host# set device-type junose
```

5. (Optional) Specify quality of service (QoS) profiles that are configured on the router.

```
[edit shared network device junose_boston]
user@host# set qos-profile [ qos-profile ...]
```

6. (Optional) Verify your configuration.

```
[edit shared network device junose_boston]
user@host# show
description "Juniper Networks E320";
management-address 10.10.8.27;
device-type junose;
qos-profile dhcp-default;
interface-classifier {
  rule rule-0 {
    script #;
  }
}
```

Adding Virtual Routers Individually (SRC CLI)

Use the following configuration statements to add a virtual router:

```
shared network device name virtual-router name {
  sae-connection [ sae-connection ...];
  snmp-read-community snmp-read-community ;
  snmp-write-community snmp-write-community ;
  scope [ scope ...];
  local-address-pools local-address-pools ;
  static-address-pools static-address-pools ;
  tracking-plug-in [ tracking-plug-in ...];
  authentication-plug-in [ authentication-plug-in ...];
  vpn-id vpn-id;
}
```

To add a virtual router:

1. From configuration mode, access the configuration statements for virtual routers. You must specify the name of a device with lowercase characters. This procedure uses `junose_Boston` as the name of the router and `vr1` as the name of the virtual router.

```
user@host# edit shared network device junose_boston virtual-router vr1
```

2. Specify the addresses of SAEs that can manage this router. This step is required for the SAE to work with the router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set sae-connection [ sae-connection ...]
```

To specify the active SAE and the redundant SAE, enter an exclamation point (!) after the hostname or IP address of the connected SAE. For example:

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set sae-connection [sae1! sae2!]
```

3. (Optional) Specify an SNMP community name for SNMP read-only operations for this VR.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set snmp-read-community snmp-read-community
```

4. (Optional) Specify an SNMP community name for SNMP write operations for this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set snmp-write-community snmp-write-community
```

5. (Optional) Specify service scopes assigned to this virtual router. The scopes are available for subscribers connected to this virtual router for selecting customized versions of services.

```
[edit shared network device junose_boston virtual-router vr1]
```

```
user@host# set scope [ scope ...]
```

6. (Optional) Specify the list of IP address pools that a virtual router currently manages and stores.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set local-address-pools local-address-pools
```

7. (Optional) Specify the list of IP address pools that a VR manages but does not store.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set static-address-pools static-address-pools
```

8. (Optional) Specify the plug-ins that track interfaces that the SAE manages on this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set tracking-plugin [ tracking-plugin ...]
user@host# set authentication-plugin [ authentication-plugin ...]
```

9. (Optional) Specify the VPN identifier used by this virtual router. For edge devices, you can specify VRF instead of a string to use the VRF instance reported by the device as the VPN identifier. For example, if you specify VRF for a JUNOS router, the VPN identifier is the name of the virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set vpn-id (vpn-id | VRF)
```

10. (Optional) Verify your configuration.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# show
sae-connection 192.168.10.25;
  snmp-read-community *****;
  snmp-write-community *****;
  scope POP-Boston;
  local-address-pools "(10.25.8.0 10.25.20.255)";
  static-address-pools "({10.30.30.0/24,10.30.30.0,10.30.30.255})";
  tracking-plugin flexRadius;
```

- Related Topics**
- Configuring the SAE to Manage JUNOS Routers (SRC CLI) on page 50
 - Specifying JUNOS Router Initialization Scripts on the SAE (SRC CLI) on page 55
 - Configuring Service Scopes (SRC CLI)
 - Types of Tracking Plug-Ins
 - Overview of Classification Scripts

Configuring the SAE to Manage JUNOS Routing Platforms (SRC CLI)

A JUNOS routing platform interacts with the SAE by using a JUNOS software process called `sdx`. When the `sdx` process establishes a TCP/IP connection to the SAE, the

SAE begins to manage the router. The JUNOS router driver configuration defines parameters related to the interactions between the SAE and the sdx process.

Use the following configuration statements to configure the JUNOS router driver:

```
shared sae configuration driver junos {  
    beep-server-port beep-server-port ;  
    tls-beep-server-port tls-beep-server-port ;  
    connection-attempts connection-attempts ;  
    keepalive-interval keepalive-interval ;  
    message-timeout message-timeout ;  
    batch-size batch-size ;  
    transaction-batch-time transaction-batch-time ;  
    sdx-group-name sdx-group-name ;  
    sdx-session-group-name sdx-session-group-name ;  
    send-commit-check send-commit-check ;  
}
```

To configure the JUNOS router driver:

1. From configuration mode, access the configuration statement that configures the JUNOS router driver. In this sample procedure, the JUNOS driver is configured in the west-region group.

```
user@host# edit shared sae group west-region configuration driver junos
```

2. Specify the TCP port number that is used to communicate with the sdx process on JUNOS routing platforms. This port number must match the port number configured in the sdx process on the router.

If you set this value to zero and the TLS BEEP server port is set, the SAE accepts only TLS connections.

```
[edit shared sae group west-region configuration driver junos]  
user@host# set beep-server-port beep-server-port
```

3. Specify the TLS port number that is used for TLS connections to the JUNOS routing platform.

If you set this value to zero, the SAE does not accept TLS connections.

```
[edit shared sae group west-region configuration driver junos]  
user@host# set tls-beep-server-port tls-beep-server-port
```

4. Specify the number of outstanding connection attempts before new connection attempts are dropped.

```
[edit shared sae group west-region configuration driver junos]  
user@host# set connection-attempts connection-attempts
```

5. Specify the interval between keepalive messages sent from the router.

```
[edit shared sae group west-region configuration driver junos]  
user@host# set keepalive-interval keepalive-interval
```

6. Specify the amount of time that the router driver waits for a response from the sdx process.

Under a high load the router may not be able to respond fast enough to requests. Change this value only if a high number of timeout events appear in the error log.

```
[edit shared sae group west-region configuration driver junos]
user@host# set message-timeout message-timeout
```

7. Specify the minimum number of service configuration transactions that are committed at the same time

```
[edit shared sae group west-region configuration driver junos]
user@host# set batch-size batch-size
```

8. Specify the maximum time to collect configuration transactions in a batch.

```
[edit shared sae group west-region configuration driver junos]
user@host# set transaction-batch-time transaction-batch-time
```

9. Specify the name of a session group on the JUNOS routing platform in which provisioning objects are stored.

```
[edit shared sae group west-region configuration driver junos]
user@host# set sdx-session-group-name sdx-session-group-name
```

10. Enable or disable commit check. If enabled, a more detailed error message is logged if a batch fails, which lets you verify individual transactions in a batch.

```
[edit shared sae group west-region configuration driver junos]
user@host# set send-commit-check send-commit-check
```

11. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration driver junos]
user@host# show
beep-server-port 3333;
tls-beep-server-port 0;
connection-attempts 50;
keepalive-interval 45;
message-timeout 30000;
batch-size 10;
transaction-batch-time 2000;
sdx-group-name sdx;
sdx-session-group-name sdx-sessions;
send-commit-check true;
```

Related Topics

- Creating Grouped Configurations for the SAE (SRC CLI)
- Configuring the SAE to Manage JUNOS Routing Platforms (C-Web Interface)
- Configuring Secure Connections Between the SAE and JUNOS Routing Platforms on page 72
- Configuring JUNOS Routing Platforms to Interact with the SAE on page 78
- Monitoring Interactions Between the SAE and JUNOS Routing Platforms on page 81

Configuring Secure Connections Between the SAE and JUNOS Routing Platforms

You can use TLS to protect communication between the SAE and JUNOS routing platforms.

To complete the handshaking protocol for the TLS connection, the client (JUNOS routing platform) and the server (SAE) must exchange and verify certificates. You need to create a client certificate and a server certificate. Both certificates must be signed by a certificate authority (CA). JUNOS software supports VeriSign, Inc. (<http://www.verisign.com>). You must then install both certificates on the SAE and on the JUNOS routing platform.

You can use SRC CLI commands to manage certificates manually, or through the Simple Certificate Enrollment Protocol (SCEP).

Certificates are in the format defined in the X.509 standard for public key infrastructure. The certificate requests are in the Public Key Cryptology Standard (PKCS) #10 format.

Tasks to set up the SAE and the JUNOS routing platform to use TLS are:

1. Adding the Server Certificate on the Routing Platform on page 72
2. Creating a Client Certificate for the Router on page 73
3. Adding the Client Certificate on the Router on page 73
4. Configuring the SAE to Use TLS (SRC CLI) on page 74
5. Configuring TLS on the SAE (SRC CLI) on page 74

- Related Topics**
- Configuring Secure Connections Between the SAE and JUNOS Routing Platforms
 - Configuring the SAE to Manage JUNOS Routing Platforms (SRC CLI) on page 69
 - BEEP Connection Between JUNOS Routing Platforms and the SAE on page 65

Adding the Server Certificate on the Routing Platform

The TLS client (JUNOS routing platform) needs a copy of the certificate that was used to sign the SAE certificate so that it can verify the SAE certificate. To install the SAE certificate on the JUNOS routing platform:

1. Include the following statements at the [edit security certificates certificate-authority] hierarchy level.

```
[edit security certificates certificate-authority]
security{
  certificates{
    certificate-authority SAE Cert{
      file /var/db/certs/cert.pem;
    }
  }
}
```


2. Include the following statements at the `[system services service-deployment]` hierarchy level.

```

system{
  services{
    service-deployment{
      servers {
        server-address port port-number{
          security-options {
            tls;
          }
        }
      }
    }
  }
}

```

- Related Topics**
- Configuring Secure Connections Between the SAE and JUNOS Routing Platforms on page 72
 - Creating a Client Certificate for the Router on page 73
 - Adding the Client Certificate on the Router on page 73
 - Configuring the SAE to Use TLS (SRC CLI) on page 74

Creating a Client Certificate for the Router

For information about how to obtain a certificate for the router from a certificate authority, see *Obtaining a Certificate from a Certificate Authority* in the *JUNOS System Basics Configuration Guide*.

- Related Topics**
- Configuring Secure Connections Between the SAE and JUNOS Routing Platforms on page 72
 - Adding the Server Certificate on the Routing Platform on page 72
 - Adding the Client Certificate on the Router on page 73

Adding the Client Certificate on the Router

To install the client (router) certificate on the JUNOS routing platform:

1. Include the following statements at the `[edit security certificates certificate-authority]` hierarchy level.

```

[edit security certificates certificate-authority]
security{
  certificates{
  }
}

```

2. Include the following statements at the **[system services service-deployment]** hierarchy level.

```
system{
  services{
    service-deployment{
      local-certificate clientCert;
    }
  }
}
```

- Related Topics**
- Configuring Secure Connections Between the SAE and JUNOS Routing Platforms on page 72
 - Adding the Server Certificate on the Routing Platform on page 72
 - Creating a Client Certificate for the Router on page 73
 - Removing a Certificate Request

Configuring the SAE to Use TLS (SRC CLI)

To configure the SAE to accept TLS connections, enter a port number with the **set beep-server-port** command in the JUNOS router driver configuration.

See “Configuring the SAE to Manage JUNOS Routing Platforms (SRC CLI)” on page 69 .

- Related Topics**
- Configuring the SAE to Use TLS
 - Configuring TLS on the SAE (SRC CLI) on page 74
 - Configuring JUNOS Routing Platforms to Interact with the SAE on page 78
 - Monitoring Interactions Between the SAE and JUNOS Routing Platforms on page 81

Configuring TLS on the SAE (SRC CLI)

Use the following configuration statements to configure TLS on the SAE:

```
shared sae configuration driver junos security {
  need-client-authentication;
  certificate-identifier private-key;
}
```

To configure TLS on the SAE:

1. From configuration mode, access the configuration statement that configures security for the JUNOS TLS connection. In this sample procedure, the JUNOS driver is configured in the west-region group.

user@host# edit shared sae group west-region configuration driver junos security
2. (Optional) Specify whether or not the SAE requests a client certificate from the router when a connection to the router is established.

```
[edit shared sae group west-region configuration driver junos security]
user@host# set need-client-authentication
```

3. Specify the name of certificate to be used for TLS communications.

```
[edit shared sae group west-region configuration driver junos security]
user@host# set certificate-identifier private-key
```

4. (Optional) Verify your TLS configuration.

```
[edit shared sae group west-region configuration driver junos security]
user@host# show
need-client-authentication;
certificate-identifier privatekey;
```

- Related Topics**
- Configuring TLS on the SAE
 - Configuring the SAE to Use TLS (SRC CLI) on page 74
 - Configuring the SAE to Manage JUNOS Routing Platforms (SRC CLI) on page 69
 - Monitoring Interactions Between the SAE and JUNOS Routing Platforms on page 81
 - BEEP Connection Between JUNOS Routing Platforms and the SAE on page 65

SAE Verification of JUNOS Configuration Changes

The SAE can check the configuration of a JUNOS routing platform under its control to detect whether the configuration has changed by a means other than through the SAE. If the SAE finds a disparity between the router and the SAE configurations, it can take several actions. The SAE checks the configuration installed on the router against the state of the SAE session layer (subscriber, service, and interface sessions). While the check is occurring, the SAE does not handle jobs from the router, and all provisioning activity is blocked, including event notifications.

The SAE can take the following actions if it finds a disparity between the router and SAE configurations:

- The SAE takes the state of the session layer on the router to be correct and updates its local state to be consistent with the router. The SAE then sends stop events for all sessions where the corresponding provisioning in the router has been removed.
- The SAE takes its local state to be the correct state and updates the router to be consistent with its local state.
- The SAE does not solve the state discrepancy. It reports disparities through the SAE device driver event trap called routerConfOutOfSynch and through the info log.

Note that it is not possible to check the consistency of individual objects that the SAE provisions. Therefore, modifications to a provisioning object while the SAE is disconnected from the router cannot be detected.

- Related Topics**
- Setting Up Periodic Configuration Checking (SRC CLI) on page 76

- [Setting Up the SAE to Periodically Check JUNOS Configuration \(C-Web Interface\)](#)

Setting Up Periodic Configuration Checking (SRC CLI)

To configure the SAE to periodically check the configuration of the JUNOS routing platform:

1. From configuration mode, access the configuration statement that configures the configuration checking feature.

```
user@host# edit shared sae configuration driver junos configuration-checking
```

2. Specify when the SAE checks the router configuration.

```
[edit shared sae configuration driver junos configuration-checking]  
user@host# set configuration-checking-schedule configuration-checking-schedule
```

3. Specify the action that the SAE takes when it detects disparities between the configuration of the SAE and the configuration on the router.

```
[edit shared sae configuration driver junos configuration-checking]  
user@host# set configuration-checking-action enforce | synchronize | detect
```

4. (Optional) From operational mode, verify your configuration checking configuration.

```
[edit shared sae configuration driver junos configuration-checking]  
user@host# show  
configuration-checking-schedule "0 0 * * * * *";  
configuration-checking-action synchronize;
```

- Related Topics**
- [Setting Up the SAE to Periodically Check JUNOS Configuration \(C-Web Interface\)](#)
 - [SAE Verification of JUNOS Configuration Changes on page 75](#)

Using SNMP to Retrieve Information from JUNOSe Routers and JUNOS Routing Platforms (SRC CLI)

You can use SNMP to retrieve information from the router. For example, if you create a router initialization script that uses SNMP, you need to specify the SNMP communities that are on the router.

We recommend that you specify SNMP communities for each virtual router. (See “Adding JUNOSe Routers and Virtual Routers (SRC CLI)” on page 46.) You can also configure global default SNMP communities.

You can configure global default SNMP communities that are used if a VR does not exist on the router or if the community strings have not been configured for the VR.

Use the following configuration statements to configure global default SNMP communities:

```
shared sae configuration driver snmp {
  read-only-community-string read-only-community-string;
  read-write-community-string read-write-community-string;
}
```

To configure global default SNMP communities:

1. From configuration mode, access the configuration statements that configure default SNMP communities.

```
user@host# edit shared sae configuration driver snmp
```

2. Configure the default SNMP community string used for read access to the router.

```
[edit shared sae configuration driver snmp]
user@host# set read-only-community-string read-only-community-string
```

3. Configure the default SNMP community string used for write access to the router.

```
[edit shared sae configuration driver snmp]
user@host# set read-write-community-string read-write-community-string
```

4. (Optional) Verify your configuration.

```
[edit shared sae configuration driver snmp]
user@host# show
read-only-community-string *****;
read-write-community-string *****;
```

- Related Topics**
- Using SNMP to Retrieve Information from JUNOS Routing Platforms and Other Network Devices
 - Configuring Event Tracking for JUNOS LSPs (SRC CLI) on page 80
 - Configuring the JUNOS Routing Platform to Apply Changes It Receives from the SAE on page 80
 - Disabling Interactions Between the SAE and JUNOS Routing Platforms on page 81
 - Monitoring Interactions Between the SAE and JUNOS Routing Platforms on page 81

Specifying Router Initialization Scripts on the SAE (SRC CLI)

Use the following configuration statements to specify router initialization scripts for JUNOS routing platforms:

```
shared sae configuration driver scripts {
  extension-path extension-path;
  general general;
  junos junos;
}
```

To configure router initialization scripts for JUNOS routing platforms:

1. From configuration mode, access the configuration statements that configure router initialization scripts. In this sample procedure, the scripts are configured in the west-region group.

```
user@host# edit shared sae group west-region configuration driver scripts
```

2. Specify the router initialization script for JUNOS routing platforms.

```
[edit shared sae group west-region configuration driver scripts]  
user@host# set junos junos
```

3. Configure a router initialization script that can be used for all types of routers that the SRC module supports.

```
[edit shared sae group west-region configuration driver scripts]  
user@host# set general general
```

4. Configure a path to router initialization scripts that are not in the default location, /opt/UMC/sae/lib.

```
[edit shared sae group west-region configuration driver scripts]  
user@host# set extension-path extension-path
```

5. (Optional) From operational mode, verify your router initialization script configuration.

```
[edit shared sae group west-region configuration driver scripts]  
user@host# show  
extension-path ;  
junos iorPublisher;
```

- Related Topics**
- Specifying JUNOS Router Initialization Scripts on the SAE (C-Web Interface)
 - Configuring JUNOS Routing Platforms to Interact with the SAE on page 78
 - Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 53

Configuring JUNOS Routing Platforms to Interact with the SAE

To configure the JUNOS routing platform to interact with the SAE:

1. Include the following statements at the **[edit system services service-deployment]** hierarchy level.

```
[edit system services service-deployment]  
servers server-address {  
  port port-number;  
}  
source-address source-address;
```

2. Use the following guidelines for the variables in these statements.

- **server-address** —Specifies the IP address of the host on which you install the SAE. Be sure this setting matches the corresponding value in the SAE configuration.
- **port-number**— Specifies the port number for the SAE. Be sure this setting matches the corresponding value in the SAE configuration.
- **source-address** —(Optional) Specifies the IP address of the source that sends traffic to the SAE.

Related Topics

- Specifying Router Initialization Scripts on the SAE (SRC CLI) on page 77
- Configuring the JUNOS Routing Platform to Apply Changes It Receives from the SAE on page 80
- Disabling Interactions Between the SAE and JUNOS Routing Platforms on page 81
- Monitoring Interactions Between the SAE and JUNOS Routing Platforms on page 81

SAE Tracking for LSPs Configured on JUNOS Routing Platforms

- Overview of SAE Tracking for LSPs Configured on JUNOS Routing Platforms on page 79
- Configuring Event Tracking for JUNOS LSPs (SRC CLI) on page 80

Overview of SAE Tracking for LSPs Configured on JUNOS Routing Platforms

You can configure the SAE to track the status of LSPs that are configured on managed JUNOS routing platforms. Use LSP tracking with applications such as the sample IPTV application. This application uses LSP tracking to collect status information for LSPs that carry IPTV traffic from video servers to a network edge router in which user connections terminate.

LSP tracking can configure the system log on managed JUNOS routing platforms to send notification messages to the managing SAE when LSPs are created and removed, and when bandwidth allocation for an LSP changes. You can enable LSP tracking for all managed JUNOS routing platforms or a set of JUNOS routing platforms.

The SAE creates a pseudointerface when each LSP becomes active (that is, when the RPD_MPLS_LSP_UP syslog event is logged) to:

- Track session status by sending interface-tracking plug-in events for each pseudointerface.
- Create subscriber sessions for the pseudo-interfaces.

The SAE does not support policy installation, including default policies, through an LSP pseudointerface.

Related Topics

- BEEP Connection Between JUNOS Routing Platforms and the SAE on page 65
- Configuring Event Tracking for JUNOS LSPs (SRC CLI) on page 80

Configuring Event Tracking for JUNOS LSPs (SRC CLI)

Configure event tracking for JUNOS LSPs to provide information to an application, such as the sample IPTV application, that needs information about LSP status.

To configure LSP tracking:

1. From configuration mode, access the configuration statement that specifies the configuration for tracking LSPs.

```
[edit]
user@host# edit shared sae configuration driver junos lsp-tracking
```
2. (Optional) Specify a regular expression to identify a set of LSP names. If you do not define an expression, the SAE tracks all LSPs.

```
[edit shared sae configuration driver junos lsp-tracking]
user@host# set match SRC123
```
3. (Optional) Specify the name of the file to store syslog event messages (that provide information about LSP state changes in a JUNOS routing platform).

For example, to store messages in the junos-1 file:

```
[edit shared sae configuration driver junos lsp-tracking]
user@host# file junos-1
```

Related Topics • Overview of SAE Tracking for LSPs Configured on JUNOS Routing Platforms on page 79

Configuring the JUNOS Routing Platform to Apply Changes It Receives from the SAE

To configure the JUNOS routing platform to receive configuration statements from the SAE and apply those statements to the configuration:

1. Create a configuration group called **sdx** that contains the configuration statements that the SAE sends to the JUNOS routing platform. To do so, include the **groups** statement at the **[edit]** level, and specify the name **sdx**.

```
[edit]
groups {
  sdx;
}
```
2. Configure the JUNOS routing platform to apply these statements to the configuration. To do so, include the **apply-groups** statement at the **[edit]** level.

```
[edit]
set apply-groups sdx;
```

Related Topics • Configuring JUNOS Routing Platforms to Interact with the SAE on page 78
• Disabling Interactions Between the SAE and JUNOS Routing Platforms on page 81
• Monitoring Interactions Between the SAE and JUNOS Routing Platforms on page 81

- Checking Changes to the JUNOS Configuration

Disabling Interactions Between the SAE and JUNOS Routing Platforms

To disable the SRC software process, enter the following command:

```
root@ui1#set system processes service-deployment disable
root@ui1# commit
```

When you disable the SRC software process, it is still available on the JUNOS routing platform.

To reenable the SRC software process, enter the following command:

```
root@ui1# delete system processes service-deployment disable
root@ui1# commit
```

The SRC software process attempts to reconnect the JUNOS routing platform to the SAE.

- Related Topics**
- Configuring JUNOS Routing Platforms to Interact with the SAE on page 78
 - Configuring the JUNOS Routing Platform to Apply Changes It Receives from the SAE on page 80
 - Monitoring Interactions Between the SAE and JUNOS Routing Platforms on page 81

Monitoring Interactions Between the SAE and JUNOS Routing Platforms

Purpose Monitor the connection between the SAE and a JUNOS routing platform.

Action Use the following command on JUNOS routing platforms to monitor the connection between the JUNOS routing platform and the SAE.

```
root@ui1>
show system services service-deployment
```

```
Connected to 172.17.20.151 port 3333 since 2004-02-06 14:50:31 PST
Keepalive settings: Interval 15 seconds
Keepalives sent: 100, Last sent: 6 seconds ago
Notifications sent: 0
Last update from peer: 00:00:06 ago
```

You can also monitor the interactions between the SRC software and JUNOS routing platforms in the log files for the SAE and in the log files generated by the SRC software process on the JUNOS routing platform.

- Related Topics**
- Configuring JUNOS Routing Platforms to Interact with the SAE on page 78
 - Configuring the JUNOS Routing Platform to Apply Changes It Receives from the SAE on page 80
 - Disabling Interactions Between the SAE and JUNOS Routing Platforms on page 81
 - Overview of Logging for SRC Components

- For information about configuring logging on JUNOS routing platforms, see *JUNOS System Basics Configuration Guide*.

Troubleshooting Problems Between the SRC module and JUNOS Device Drivers

- Troubleshooting Problems with the SRC Software Process on page 82
- Viewing the State of JUNOS Device Drivers (SRC CLI) on page 83
- Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI) on page 83
- Viewing Statistics for All JUNOS Device Drivers (SRC CLI) on page 84
- Viewing the State of JUNOS Device Drivers (C-Web Interface) on page 85
- Viewing Statistics for Specific JUNOS Device Drivers (C-Web Interface) on page 86
- Viewing Statistics for All JUNOS Device Drivers (C-Web Interface) on page 86

Troubleshooting Problems with the SRC Software Process

Problem The SRC process on a JUNOS routing platform is not working as expected.

Solution Review the log files for the SAE and the log files generated by the SRC software process on the router. If the log files indicate that the SRC software process on the JUNOS routing platform is not responding:

1. Look at the status of the process on the JUNOS routing platform.

```
root@ui1>show system services service-deployment
```

```
Connected to 172.17.20.151 port 3333 since 2004-02-06 14:50:31 PST
Keepalive settings: Interval 15 seconds
Keepalives sent: 100, Last sent: 6 seconds ago
Notifications sent: 0
Last update from peer: 00:00:06 ago
```

2. If you see the message "error: the service-deployment subsystem is not running," reenable the SRC software process. See "Disabling Interactions Between the SAE and JUNOS Routing Platforms" on page 81.
3. If the process is already enabled, review the configurations of the router and the SAE in the directory, and fix any problems.
4. Restart the SRC software process on the router.

```
root@ui1>restart service-deployment
```

The SAE synchronizes with the SRC software process and deletes unnecessary data from the router.

If deleting parts of the SRC data on a JUNOS routing platform fails to solve problems, delete all the SRC data and restart the SRC software process. To do so:

1. Delete all SRC interfaces and services.

```
delete groups sdx
root@ui1#commit
```

- Restart the SRC software process on the router.

```
root@ui1>restart service-deployment
```

- Related Topics**
- Viewing the State of JUNOS Device Drivers (SRC CLI) on page 83
 - Viewing Statistics for All JUNOS Device Drivers (SRC CLI) on page 84

Viewing the State of JUNOS Device Drivers (SRC CLI)

Purpose Display the state of JUNOS drivers.

Action Use the following operational mode command:

```
show sae drivers <device-name device-name > <(brief) > <maximum-results  
maximum-results >
```

For example:

```
user@host> show sae drivers device-name default@jrouter
JUNOS Driver
Device name                default@jrouter
Device type                junos
Device IP                  /10.10.6.113:1879
Local IP                   10.10.6.113
TransportRouter
Device version              8.2R1.7
Start time                  Thu Mar 08 21:00:50 UTC 2007
Number of notifications     0
Number of processed added   0
Number of processed changed 0
Number of processed deleted 0
Number of provisioning attempt 0
Number of provisioning attempt failed 0
Device type                JunosRouterDriver
Job queue size              0
Number of SAP                3
Number of PAP                0
Start time                  Thu Mar 08 21:00:55 UTC 2007
End time                    Thu Mar 08 21:00:55 UTC 2007
Transaction Manager
Transaction queue size 0
Router name                 default@tro11
```

- Related Topics**
- Viewing the State of JUNOS Device Drivers (C-Web Interface) on page 85
 - Troubleshooting Problems with the SRC Software Process on page 82
 - Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI) on page 83
 - Viewing Statistics for All JUNOS Device Drivers (SRC CLI) on page 84

Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI)

Purpose Display statistics for a specific JUNOS device driver.

Action Use the following operational mode command:

`show sae statistics device <name name> <(brief) >`

For example:

```
user@host> show sae statistics device name default@jrouter
SNMP Statistics
Add notification handle time      7
Change notification handle time   0
Client ID                        default@troll
Delete notification handle time   0
Failover IP                      0.0.0.0
Failover port                    0
Handle message time              40
Job queue age                    0
Job queue time                   0
Number message send               3
Number of added jobs              0
Number of add notifications       0
Number of change notifications    0
Number of delete notifications    0
Number of managed interfaces      3
Number of message errors          0
Number of message timeouts        0
Number of removed jobs            0
Number of user session established 0
Number of user session removed    0
Router type                      JUNOS
Up time                          7036120
Using failover server             false
```

- Related Topics**
- Viewing Statistics for Specific JUNOS Device Drivers (C-Web Interface) on page 86
 - Viewing the State of JUNOS Device Drivers (SRC CLI) on page 83
 - Viewing Statistics for All JUNOS Device Drivers (SRC CLI) on page 84

Viewing Statistics for All JUNOS Device Drivers (SRC CLI)

Purpose Display SNMP statistics for all JUNOS device drivers.

Action Use the following operational mode command:

`show sae statistics device common junos`

For example:

```
user@host> show sae statistics device common junos
SNMP Statistics
Driver type                      JUNOS
Number of close requests         0
Number of connections accepted   0
Number of current connections    0
Number of open requests          0
Server address                   0.0.0.0
Server port                      3288
Time since last redirect         0
```

- Related Topics**
- Viewing Statistics for All JUNOS Device Drivers (C-Web Interface) on page 86
 - Viewing the State of JUNOS Device Drivers (SRC CLI) on page 83

- Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI) on page 83

Viewing the State of JUNOS Device Drivers (C-Web Interface)

Problem Log files indicate a problem with a specific driver.

Solution Review the configuration of the associated JUNOS router driver with C-Web:

1. Select **SAE** from the side pane, and click **Drivers**.

The Drivers pane appears.

The screenshot shows the Juniper C-Web interface. On the left is a sidebar with a menu including Monitor, ACP, CLI, Component, Date, Disk, Interfaces..., JPS, NIC, NTP, Redirect Server, Route..., SAE (highlighted), Security, and System. The main content area is titled 'SAE Drivers'. It contains three configuration fields: 'Name Of Device Driver' with a text input box, 'Style' with a dropdown menu, and 'Maximum Results' with a text input box. To the right of these fields are help text and choices: 'Name of device drivers. Please enter: All or part of the device driver name. For JUNOS router drivers and PCMM drivers, use the format default@routerName.', 'Output style Choices: brief: Display only virtual router names', and 'Number of results to be displayed. Legal range: 1 .. INF. Default value: 25'. At the bottom of the configuration area are 'OK' and 'Reset' buttons. The footer of the interface shows 'Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy.' and the Juniper logo with the tagline 'Juniper your Net.'

2. In the Name of Device Driver box, enter a full or partial device driver name for which you want to display information, or leave the box blank to display all devices. Use the format:
default@<router name>
3. Select an output style from the Style list.
4. In the Maximum Results box, enter the maximum number of results that you want to receive.
5. Click **OK**.

The Drivers pane displays information about the JUNOS device driver.

- Related Topics**
- Viewing the State of JUNOS Device Drivers (SRC CLI) on page 83
 - Viewing Statistics for Specific JUNOS Device Drivers (C-Web Interface) on page 86
 - Viewing Statistics for All JUNOS Device Drivers (C-Web Interface) on page 86

Viewing Statistics for Specific JUNOS Device Drivers (C-Web Interface)

Purpose View SNMP statistics about devices.

- Action**
1. Select **SAE** from the side pane, click **Statistics**, and then click **Device**.
The Device pane appears.

The screenshot shows the Juniper C-Web Interface. The top navigation bar includes 'Monitor', 'Logged in as: admin', and links for 'About', 'Refresh', and 'Logout'. The left sidebar lists components: ACP, CLI, Component, Date, Disk, Interfaces..., JPS, NIC, NTP, Redirect Server, Route..., SAE (highlighted), Security, and System. The main content area is titled 'SAE Device'. It contains a form with a 'Device Name' text box and a 'Style' dropdown menu. Below the form are 'OK' and 'Reset' buttons. To the right of the form, there is a help text: 'Name of a device. Please enter: All or part of the device name. For JUNOS router drivers and PCMM drivers, use the format default@routerName.' and 'Output style Choices: brief: Display only device names'. The bottom of the page shows the copyright notice: 'Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy.' and the Juniper logo.

2. In the Device Name box, enter a full or partial device name for which you want to display information, or leave the box blank to display all devices.
3. Select an output style from the Style list.
4. Click **OK**.

The Device pane displays statistics for all devices.

- Related Topics**
- Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI) on page 83
 - Viewing the State of JUNOS Device Drivers (C-Web Interface) on page 85
 - Viewing Statistics for All JUNOS Device Drivers (C-Web Interface) on page 86

Viewing Statistics for All JUNOS Device Drivers (C-Web Interface)

Purpose View SNMP statistics about specific devices.

- Action** 1. Select **SAE** from the side pane, click **Statistics**, click **Device**, and then click **Common**.
The Common pane appears.

The screenshot shows the Juniper SAE web interface. On the left is a navigation pane with links: Monitor, ACP, CLI, Component, Date, Disk, Interfaces..., JPS, NIC, NTP, Redirect Server, Route..., SAE (highlighted), Security, and System. The main content area is titled 'SAE' and 'Common'. It contains two input fields: 'Device Name' (a text box) and 'Type' (a dropdown menu). Below these fields are 'OK' and 'Reset' buttons. To the right of the 'Type' dropdown is a text area containing the following information:

Name of a device.
Please enter: All or part of the device name. For JUNOS router drivers and PCMM drivers, use the format default@routerName.

Display SNMP statistics for a specified device driver type.
Choices:
junos: Display SNMP statistics for JUNOS router drivers
junose-cops: Display SNMP statistics for JUNOS router drivers
packetcable-cops: Display SNMP statistics for PCMM device drivers
proxy: Display SNMP statistics for third-party drivers

At the bottom of the interface, there is a copyright notice: 'Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy.' and the Juniper logo with the tagline 'Juniper Your Net.'

2. In the Device Name box, enter a full or partial device name for which you want to display information, or leave the box blank to display all devices.
3. Select the **junos**JUNOS from the Type list:
4. Click **OK**.

The Common pane displays statistics for the specified device.

- Related Topics**
- Viewing Statistics for All JUNOS Device Drivers (SRC CLI) on page 84
 - Viewing the State of JUNOS Device Drivers (C-Web Interface) on page 85
 - Viewing Statistics for Specific JUNOS Device Drivers (C-Web Interface) on page 86

PART 3

Using Network Devices in the SRC Network

- Integrating Third-Party Network Devices into the SRC Network (SRC CLI) on page 91

CHAPTER 8

Integrating Third-Party Network Devices into the SRC Network (SRC CLI)

- Overview of Integrating Network Devices into the SRC Network on page 91
- Logging In Subscribers and Creating Sessions on page 93
- Configuration Tasks for Integrating Third-Party Network Devices (SRC CLI) on page 96
- Setting Up Script Services on page 97
- Adding Objects for Network Devices (SRC CLI) on page 98
- Adding Virtual Router Objects (SRC CLI) on page 99
- Setting Up SAE Communities (SRC CLI) on page 100
- Configuring SAE Properties for the Event Notification API (SRC CLI) on page 102
- Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 103
- Copying Initialization Scripts to the C Series Controller on page 106
- Specifying Initialization Scripts on the SAE (SRC CLI) on page 106
- Using SNMP to Retrieve Information from Network Devices on page 107
- Configuring Global SNMP Communities in the SRC Software (SRC CLI) on page 107
- Using the NIC Resolver in Environments That Have Third-Party Devices (SRC CLI) on page 108

Overview of Integrating Network Devices into the SRC Network

You can integrate third-party routers and other network devices into your SRC network. The SAE provides a driver that you can use to integrate the SAE with a third-party device. This device driver uses the session store to store and replicate subscriber and service session data within a community of SAEs.

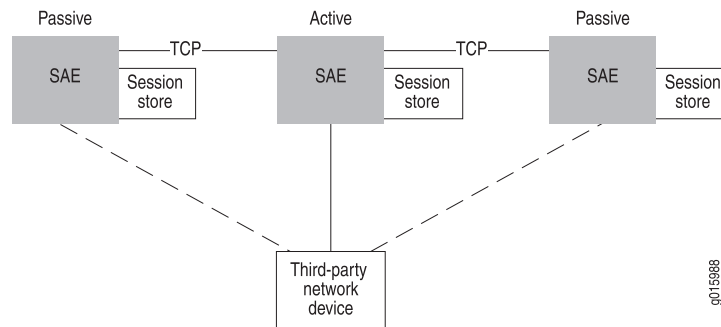
To log in subscribers to the SAE, you use assigned IP subscribers or event notification from an IP address manager.

To activate services and provision policies on the device, you use script services. You can also activate aggregate services for subscribers. However, you cannot activate normal services that require policies to be provisioned on the device.

SAE Communities

For SAE redundancy in an SRC network, you can have a community of two or more SAEs. SAEs in a community are given the role of either active SAE or passive SAE. The active SAE manages the connection to the network device and keeps session data up to date within the community. Figure 7 on page 92 shows a typical SAE community.

Figure 7: SAE Community



When an SAE starts, it negotiates with other SAEs to determine which SAE controls the network device. The SAE community manager and members of the community select the active SAE.

A passive SAE needs to take over as active SAE in any of the following cases:

- The active SAE shuts down. In this case, the active SAE notifies the passive SAEs, and one of the passive SAEs takes over as active SAE.
- A passive SAE does not receive a keepalive message from the active SAE within the keepalive interval. In this case, the passive SAE attempts to become the active SAE.

Storing Session Data

To aid in recovering from an SAE failover, the SAE stores subscriber and service session data. When the SAE manages a network device, session data is stored in the SAE host's file system. The SRC component that controls the storage of session data on the SAE is called the session store. The session store queues data and then writes the data to session store files on the SAE host's disk. Once the data is written to disk, it can survive a server reboot.

For more information, see "Storing Subscriber and Service Session Data" on page 23.

Using Script Services to Provision Third-Party Devices

You use script services to activate services and provision policies on third-party network devices. A script service is a service into which you can insert or reference a script. You write a script that will activate services and provision policies on the third-party device, and then you insert the script into the script service or reference the script in the service. When the SAE activates a service, it runs the script. The script provisions policies on the device using a means that the device supports. You can also include an interface in the script that causes the SAE to send authentication and tracking events when it activates, modifies, or deactivates a script service session.

The SAE core API includes two interfaces for creating a script:

- **ScriptService**—Defines a service provider interface (SPI) that the script service must implement. The implementation of the ScriptService interface activates, modifies, or deactivates the service.
- **ServiceSessionInfo**—Provides a callback interface into the SAE and provides information about the service session to the script service.

For information about the ScriptService interface and the ServiceSessionInfo interface, see the script service documentation in the SAE core API documentation on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/src/api-index.html>

You can write the script in Java or Jython.

- Related Topics**
- Logging In Subscribers and Creating Sessions on page 93
 - Configuration Tasks for Integrating Third-Party Network Devices (SRC CLI) on page 96
 - Adding Objects for Network Devices (C-Web Interface)
 - Setting Up SAE Communities (C-Web Interface)
 - Configuring the SAE Community Manager

Logging In Subscribers and Creating Sessions

You can use two mechanisms to obtain subscriber address requests and other information and to set up a pseudointerface on the network device. (You must choose one mechanism; you cannot mix them.)

1. **Assigned IP subscriber.** The SAE learns about a subscriber through subscriber-initiated activities, such as activating a service through the portal or through the SRC Web Services Gateway).

With this method, you use the assigned IP subscriber login type along with the network interface collector (NIC) to map IP addresses to the SAE.

2. **Event notification from an IP address manager.** The SAE learns about subscribers through notifications from an external IP address manager, such as a DHCP server or a RADIUS server.

With this method, you use the event notification application programming interface (API). The API provides an interface to the IP address manager, and lets the IP address manager notify the SAE of events such as IP address assignments.

Assigned IP Subscribers

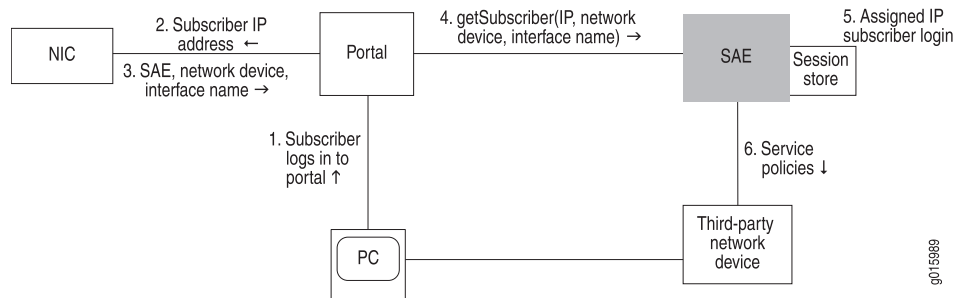
With the assigned IP subscriber method of logging in subscribers and creating sessions, the SRC module uses IP address pools along with network information collector (NIC) resolvers to provide mapping of IP addresses to SAEs. You configure the static address pools or dynamically discovered address pools in the virtual router configuration for a network device. These pools are published in the NIC. The NIC maps subscriber IP

addresses in requests received through the portal or SRC Web Services Gateway to the SAE that currently manages that network device.

Login Interactions with Assigned IP Subscribers

This section describes login interactions for assigned IP subscribers. In the example shown in Figure 8 on page 94, the subscriber activates a service through a portal. You could also have the subscriber activate a service through the SRC Web Services Gateway.

Figure 8: Login Interactions with Assigned IP Subscribers



The sequence of events for logging in and creating sessions for assigned IP subscribers is:

1. The subscriber logs in to the portal.
2. The portal sends the subscriber's IP address to the NIC.
3. Based on the IP address, the NIC looks up the subscriber's SAE, network device, and interface name, and returns this information to the portal.
4. The portal sends a get Subscriber message to the SAE. The message includes the subscriber's IP address, network device, and interface name.
5. The SAE creates an assigned IP subscriber and performs a subscriber login. Specifically, it:
 - a. Runs the subscriber classification script with the IP address of the subscriber. (Use the ASSIGNEDIP login type in subscriber classification scripts.)
 - b. Loads the subscriber profile.
 - c. Runs the subscriber authorization plug-ins.
 - d. Runs the subscriber tracking plug-ins.
 - e. Creates a subscriber session and stores the session data in the session store file.
6. The SAE pushes service policies for the subscriber session to the network device.

Because the SAE is not notified when the subscriber logs out, the assigned IP idle timer begins when no service is active. The SAE removes the interface subscriber session when the timeout period ends.

Event Notification from an IP Address Manager

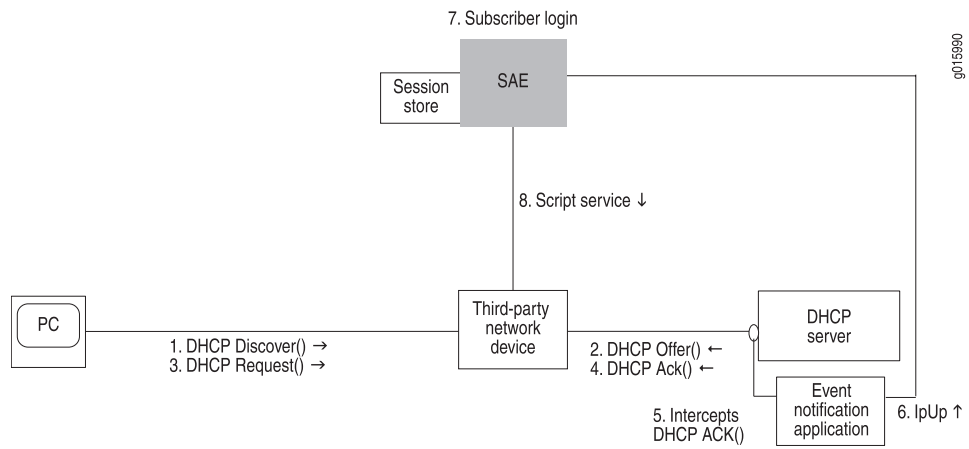
With the event notification method of logging in subscribers and creating subscriber sessions, the subscriber logs in to the network device and obtains an IP address through an address server, usually a DHCP server. The SAE receives notifications about the subscriber, such as the subscriber's IP address, from an event notification application that is installed on the DHCP server.

To use this method of logging in subscribers, you can use the event notification API to create the application that notifies the SAE when events occur between the DHCP server and the network device. You can also use Monitoring Agent, a sample application that was created with the event notification API and that monitors DHCP or RADIUS messages for DHCP or RADIUS servers. See the *SRC PE Sample Applications Guide*.

Login with Event Notification

This section describes login interactions by means of event notifications.

Figure 9: Login Interactions with Event Notification Application



The sequence of events for logging in subscribers and creating sessions is:

1. The DHCP client in the subscriber's computer sends a DHCP discover request to the DHCP server.
2. The DHCP server sends a DHCP offer to the subscriber's DHCP client.
3. The DHCP client sends a DHCP request to the DHCP server.
4. The DHCP server acknowledges the request by sending a DHCP Ack message to the DHCP client.
5. The event notification application that is running on the DHCP server intercepts the DHCP Ack message.
6. The event notification application sends an ipUp message to the SAE that notifies the SAE that an IP address is up.
7. The SAE performs a subscriber login. Specifically, it:

- a. Runs the subscriber classification script.
 - b. Loads the subscriber profile.
 - c. Runs the subscriber authorization plug-ins.
 - d. Runs the subscriber tracking plug-ins.
 - e. Creates a subscriber session and stores the session in the session store file.
8. The SAE can start script services.

The ipUp event should be sent with a timeout set to the DHCP lease time. The DHCP server sends an ipUp event for each Ack message sent to the client. The SAE restarts the timeout each time it receives an ipUp event.

If the client explicitly releases the DHCP address (that is, it sends a DHCP release event), the DHCP server sends an ipDown event. If the client does not renew the address, the lease expires on the DHCP server and the timeout expires on the SAE.

- Related Topics**
- Overview of Integrating Network Devices into the SRC Network on page 91
 - Using the NIC Resolver in Environments That Have Third-Party Devices (C-Web Interface)
 - Configuration Tasks for Integrating Third-Party Network Devices (SRC CLI) on page 96
 - Configuring SAE Properties for the Event Notification API (SRC CLI) on page 102
 - Adding Objects for Network Devices (SRC CLI) on page 98
 - Setting Up Script Services on page 97

Configuration Tasks for Integrating Third-Party Network Devices (SRC CLI)

To integrate third-party devices into your SRC network, complete the following tasks:

- Write a script and add a script service that references the script.
See “Setting Up Script Services” on page 97.
- Add objects for the devices.
See “Adding Objects for Network Devices (SRC CLI)” on page 98.
- Configure an SAE community.
See “Setting Up SAE Communities (SRC CLI)” on page 100.
- (Optional) Configure SAE properties for the Event Notification API if you are using the event notification method to log in subscribers.
See “Configuring SAE Properties for the Event Notification API (SRC CLI)” on page 102.
- Configure the session store.

See “Storing Subscriber and Service Session Data” on page 23.

- If you are using the event notification method to log in subscribers, integrate the SAE with an IP address manager. There are two ways to do so:
 - Use the event notification API to create an application that notifies the SAE when events occur between the DHCP server and the network device.

See the event notification API documentation in the SAE CORBA remote API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>
 - Use Monitoring Agent, a sample application that was created with the event notification API and that monitors DHCP or RADIUS messages for DHCP or RADIUS servers.

See the *SRC PE Sample Applications Guide*.

- Related Topics**
- Configuration Tasks for Integrating Third-Party Network Devices (C-Web Interface)
 - Overview of Integrating Network Devices into the SRC Network on page 91
 - Logging In Subscribers and Creating Sessions on page 93

Setting Up Script Services

To set up script services:

1. Write a script that implements the ScriptService interface, a service provider interface (SPI) for the SAE.

See Customizing Service Implementations.

See the script service documentation in the SAE core API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>
2. Add a script service that references the script.

See Overview of SRC Script Services.

- Related Topics**
- Configuration Tasks for Integrating Third-Party Network Devices (SRC CLI) on page 96
 - Copying Initialization Scripts to the C Series Controller on page 106
 - Overview of Integrating Network Devices into the SRC Network on page 91
 - Logging In Subscribers and Creating Sessions on page 93
 - Setting Up SAE Communities (C-Web Interface)

Adding Objects for Network Devices (SRC CLI)

For each network device that the SAE manages, add a router object and virtual router object.

Use the following configuration statements to add a router object:

```
shared network device name {  
  description description;  
  management-address management-address;  
  device-type (junose| junos| pcmm| third-party);  
  qos-profile [qos-profile...];  
}
```

To add a router object:

1. From configuration mode, access the statements that configure network devices. You must specify the name of a device with lowercase characters. This sample procedure uses proxy_device as the name of the router.

```
user@host# edit shared network device proxy_device
```

2. (Optional) Add a description for the router object.

```
[edit shared network device proxy_device]  
user@host# set description description
```

3. (Optional) Add the IP address of the router object.

```
[edit shared network device proxy_device]  
user@host# set management-address management-address
```

4. Set the type of device that you are adding to third-party.

```
[edit shared network device proxy_device]  
user@host# set device-type third-party
```

5. (Optional) Verify your configuration.

```
[edit shared network device proxy_device]  
user@host# show  
description "Third-party router";  
management-address 192.168.9.25;  
device-type third-party;  
interface-classifier {  
  rule rule-0 {  
    script #;  
  }  
}
```

Related Topics

- Adding Objects for Network Devices (C-Web Interface)
- Configuration Tasks for Integrating Third-Party Network Devices (SRC CLI) on page 96
- Adding Virtual Router Objects (SRC CLI) on page 99

- Overview of Integrating Network Devices into the SRC Network on page 91
- Logging In Subscribers and Creating Sessions on page 93

Adding Virtual Router Objects (SRC CLI)

Use the following configuration statements to add a virtual router:

```
shared network device name virtual-router name {
  sae-connection [sae-connection ...];
  snmp-read-community snmp-read-community;
  snmp-write-community snmp-write-community;
  scope [scope...];
  tracking-plug-in [tracking-plug-in...];
}
```

To add a virtual router:

1. From configuration mode, access the statements for virtual routers. You must specify the name of a device with lowercase characters. This sample procedure uses `proxy_device` as the name of the router object. For third-party devices, use the name default for the virtual router.

```
user@host# edit shared network device proxy_device virtual-router default
```

2. Specify the addresses of SAEs that can manage this router. This step is required for the SAE to work with the router.

```
[edit shared network device proxy_device virtual-router default]
user@host# set sae-connection [sae-connection ...]
```

To specify the active SAE and the redundant SAE, enter an exclamation point (!) after the hostname or IP address of the connected SAE. For example:

```
[edit shared network device proxy_device virtual-router default]
user@host# set sae-connection [sae1! sae2!]
```

3. (Optional) Specify an SNMP community name for SNMP read-only operations for this virtual router.

```
[edit shared network device proxy_device virtual-router default]
user@host# set snmp-read-community snmp-read-community
```

4. (Optional) Specify an SNMP community name for SNMP write operations for this virtual router.

```
[edit shared network device proxy_device virtual-router default]
user@host# set snmp-write-community snmp-write-community
```

5. (Optional) Specify service scopes assigned to this virtual router. The scopes are available for subscribers connected to this virtual router for selecting customized versions of services.

```
[edit shared network device proxy_device virtual-router default]
```

```
user@host# set scope [ scope ...]
```

6. (Optional) Specify the plug-ins that track interfaces that the SAE manages on this virtual router.

```
[edit shared network device proxy_device virtual-router default]  
user@host# set tracking-plugin [ tracking-plugin ...]
```

7. (Optional) Verify your configuration.

```
[edit shared network device proxy_device virtual-router default]  
user@host# show  
sae-connection 10.8.221.45;  
snmp-read-community *****;  
snmp-write-community *****;  
scope POP-Toronto;  
tracking-plugin flexRadius;
```

- Related Topics**
- Adding Objects for Network Devices (C-Web Interface)
 - Adding Objects for Network Devices (SRC CLI) on page 98
 - Overview of Integrating Network Devices into the SRC Network on page 91

Setting Up SAE Communities (SRC CLI)

Tasks to configure SAE communities are:

- If there is a firewall in the network, configuring the firewall to allow SAE messages through.
1. Configuring the SAE Community Manager on page 100
 2. Specifying the Community Manager in the SAE Device Driver on page 101

Configuring the SAE Community Manager

Use the following configuration statements to configure the SAE community manager that manages third-party network device communities:

```
shared sae configuration external-interface-features name CommunityManager {  
  keepalive-interval keepalive-interval ;  
  threads threads ;  
  acquire-timeout acquire-timeout ;  
  blackout-time blackout-time ;  
}
```

To configure the community manager:

1. From configuration mode, access the configuration statements for the community manager. In this sample procedure, *sae_mgr* is the name of the community manager.

```
user@host# edit shared sae configuration external-interface-features sae_mgr  
CommunityManager
```

- Specify the interval between keepalive messages sent from the active SAE to the passive members of the community.

```
[edit shared sae configuration external-interface-features sae_mgr
CommunityManager]
user@host# set keepalive-interval keepalive-interval
```

- Specify the number of threads that are allocated to manage the community. You generally do not need to change this value.

```
[edit shared sae configuration external-interface-features sae_mgr
CommunityManager]
user@host# set threads threads
```

- Specify the amount of time an SAE waits for a remote member of the community when it is acquiring a distributed lock. You generally do not need to change this value.

```
[edit shared sae configuration external-interface-features sae_mgr
CommunityManager]
user@host# set acquire-timeout acquire-timeout
```

- Specify the amount of time that an active SAE must wait after it shuts down before it can try to become the active SAE of the community again.

```
[edit shared sae configuration external-interface-features sae_mgr
CommunityManager]
user@host# set blackout-time blackout-time
```

- (Optional) Verify the configuration of the SAE community manager.

```
[edit shared sae configuration external-interface-features sae_mgr
CommunityManager]
user@host# show
CommunityManager {
  keepalive-interval 30;
  threads 5;
  acquire-timeout 15;
  blackout-time 30;
}
```

Specifying the Community Manager in the SAE Device Driver

Use the following configuration statements to specify the community manager in the SAE device driver.

```
shared sae configuration driver third-party {
  sae-community-manager sae-community-manager;
}
```

To specify the community manager:

- From configuration mode, access the configuration statements for the third-party device driver.

```
user@host# edit shared sae configuration driver third-party
```

2. Specify the name of the community manager.

```
[edit shared sae configuration driver third-party]
user@host# set sae-community-manager sae-community-manager
```

3. (Optional) Verify the configuration of the third-party device driver.

```
[edit shared sae configuration driver third-party]
user@host# show
sae-community-manager sae_mgr;
```

Related Topics

- Setting Up SAE Communities (C-Web Interface)
- Configuration Tasks for Integrating Third-Party Network Devices (SRC CLI) on page 96
- Configuring the SAE Community Manager
- Overview of Integrating Network Devices into the SRC Network on page 91
- Logging In Subscribers and Creating Sessions on page 93

Configuring SAE Properties for the Event Notification API (SRC CLI)

Use the following configuration statements to configure properties for the Event Notification API:

```
shared sae configuration external-interface-features name EventAPI {
  retry-time retry-time ;
  retry-limit retry-limit ;
  threads threads ;
}
```

To configure properties for the Event Notification API:

1. From configuration mode, access the configuration statements for the Event Notification API. In this sample procedure, west-region is the name of the SAE group, and event_api is the name of the Event API configuration.

```
user@host# edit shared sae group west-region configuration
external-interface-features event_api EventAPI
```

2. Specify the amount of time between attempts to send events that could not be delivered.

```
[edit shared sae group west-region configuration external-interface-features event_api
EventAPI]
user@host# set retry-time retry-time
```

3. Specify the number of times an event fails to be delivered before the event is discarded.

```
[edit shared sae group west-region configuration external-interface-features event_api
EventAPI]
user@host# set retry-limit retry-limit
```

4. Specify the number of threads allocated to process events.

```
[edit shared sae group west-region configuration external-interface-features event_api
EventAPI]
user@host# set threads threads
```

5. (Optional) Verify the configuration of the Event Notification API properties.

```
[edit shared sae group west-region configuration
external-interface-features event_api EventAPI]
user@host# show
EventAPI {
  retry-time 300;
  retry-limit 5;
  threads 5;
}
```

- Related Topics**
- Using the SAE in a PCMM Environment
 - Configuring SAE Properties for the Event Notification API (C-Web Interface)
 - Initially Configuring the SAE
 - Configuring the SAE to Manage PCMM Devices (SRC CLI)

Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers

When the SAE establishes a connection with a router or network device, it can run an initialization script to customize the setup of the connection. These initialization scripts are run when the connection between a router or network device and the SAE is established and again when the connection is dropped.

We provide the `lorPublisher` script in the `/opt/UMC/sae/lib` folder. The `lorPublisher` script publishes the interoperable object reference (IOR) of the SAE in the directory so that a NIC can associate a router with an SAE.

For JUNOSe VRs that supply IP addresses from a local pool, a router initialization script is provided that identifies which VR supplies each IP pool and writes the information to the configuration. The SAE runs the script only when a COPS connection is established to the JUNOSe router. Consequently, if you modify information about IP pools on a VR after the COPS connection is established, the SAE will not automatically register the changes, and you must update the configuration.

Table 3 on page 53 describes the router initialization scripts that we provide with the SRC software in the `/opt/UMC/sae/lib` folder.

Table 5: Router Initialization Scripts

Script Name	Function	When to Use Script
iorPublisher	Publishes the IOR of the SAE into an internal part of the shared configuration so that a NIC can associate a router with an SAE.	Use with JUNOS routers that do not supply IP addresses from local pools, and with JUNOS routing platforms. Use with all JUNOS routing platforms. Use with third-party network devices.
poolPublisher	Publishes the IOR of the SAE and local IP address pools in the directory so that a NIC can associate a router with an SAE and resolve the IP-to-SAE mapping.	Use with JUNOS virtual routers that supply IP addresses from local pools.

Interface Object Fields

Router initialization scripts are written in the Python programming language (www.python.org) and executed in the Jython environment (www.jython.org).

Router initialization scripts interact with the SAE through an interface object called Ssp. The SAE exports a number of fields through the interface object to the script and expects the script to provide the entry point to the SAE.

Table 4 on page 54 describes the fields that the SAE exports.

Table 6: Exported Fields

Ssp Attribute	Description
Ssp.properties	System properties object (class: java.util.Properties)—The properties should be treated as read-only by the script.
Ssp.errorLog	Error logger—Use the Ssp.errorLog.println (message) to send error messages to the log.
Ssp.infoLog	Info logger—Use the Ssp.infoLog.println (message) to send informational messages to the log.
Ssp.debugLog	Debug logger—Use the Ssp.debugLog.println (message) to send debug messages to the log.

The router initialization script must set the field Ssp.routerInit to a factory function that instantiates a router initialization object:

- <VRName>—Name of the virtual router in which the COPS client has been configured, format: virtualRouterName@RouterName
- <virtualIp>—Virtual IP address of the SAE (string, dotted decimal; for example: 192.168.254.1)
- <realIp>—Real IP address of the SAE (string, dotted decimal; for example, 192.168.1.20)

- <VRip>—IP address of the virtual router (string, dotted decimal)
- <transportVR>—Name of the virtual router used for routing the COPS connection, or None, if the COPS client is directly connected

The factory function must implement the following interface:

```
Ssp.routerInit(VRName,
virtualIp,
realIp,
VRip,
transportVR)
```

The factory function returns an interface object that is used to set up and tear down a connection for a given COPS server. A common case of a factory function is the constructor of a class.

The factory function is called directly after a COPS server connection is established. In case of problems, an exception should be raised that leads to the termination of the COPS connection.

Required Methods

Instances of the interface object must implement the following methods:

- *setup()*—Is called when the COPS server connection is established and is operational. In case of problems, an exception should be raised that leads to the termination of the COPS connection.
- *shutdown()*—Is called when the COPS server connection to the virtual router is terminated. This method should not raise any exceptions in case of problems.

Example: Router Initialization Script

The following script defines a router initialization class named *SillyRouterInit*. The interface class does not implement any useful functionality. The interface class just writes messages to the infoLog when the router connection is created or terminated.

```
class SillyRouterInit:
    def __init__(self, vrName, virtualIp, realIp, vrIp, transportVr):
        """ initialize router initialization object """
        self.vrName = vrName
        Ssp.infoLog.println("SillyRouterInit created")
    def setup(self):
        """ initialize connection to router """
        Ssp.infoLog.println("Setup connection to VR %(vrName)s" %
            vars(self))
    def shutdown(self):
        """ shutdown connection to router """
        Ssp.infoLog.println("Shutdown connection to VR %(vrName)s" %
            vars(self))
#
# publish interface object to Ssp core
#
Ssp.routerInit = SillyRouterInit
```

Related Topics • How SNMP Obtains Information from Routers for the SRC Software on page 52

- Specifying JUNOS Router Initialization Scripts on the SAE (SRC CLI) on page 55
- Accessing the Router CLI on page 57
- Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI) on page 61
- Troubleshooting Problems with Managing JUNOS Routers on page 59

Copying Initialization Scripts to the C Series Controller

If you use a script that is not provided with the SRC module, you need to use the **file copy** command to copy your script to the C Series Controller. For example:

```
user@host> file copy ftp://user@myserver/routerinit.py /opt/UMC/sae/lib
Password:
```

- Related Topics**
- Specifying Initialization Scripts on the SAE (SRC CLI) on page 106
 - Setting Up Script Services on page 97
 - Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 53

Specifying Initialization Scripts on the SAE (SRC CLI)

Use the following configuration statements to specify initialization scripts for third-party devices:

```
shared sae configuration driver scripts {
  extension-path extension-path;
  general general;
}
```

To configure initialization scripts for third-party devices:

1. From configuration mode, access the configuration statements that configure initialization scripts.

```
user@host# edit shared sae configuration driver scripts
```
2. Specify the initialization script for third-party devices.

```
[edit shared sae configuration driver scripts]
user@host# set general general
```
3. Configure a path to scripts that are not in the default location, `/opt/UMC/sae/lib`.

```
[edit shared sae configuration driver scripts]
user@host# set extension-path extension-path
```
4. (Optional) Verify your initialization script configuration.

```
[edit shared sae configuration driver scripts]
user@host# show
```

- Related Topics**
- Specifying Initialization Scripts on the SAE (C-Web Interface)
 - Copying Initialization Scripts to the C Series Controller on page 106
 - Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 53

Using SNMP to Retrieve Information from Network Devices

You can use SNMP to retrieve information from a network device. For example, if you create a script that uses SNMP, specify the SNMP communities that are on the network device.

To retrieve information:

- (Recommended) Specify SNMP communities for each virtual router object.
- Configure global default SNMP communities.

- Related Topics**
- Adding Virtual Router Objects (SRC CLI) on page 99
 - Adding Objects for Network Devices (C-Web Interface)
 - Configuring Global SNMP Communities in the SRC Software (SRC CLI) on page 107
 - Configuring Global SNMP Communities in the SRC Software (C-Web Interface)

Configuring Global SNMP Communities in the SRC Software (SRC CLI)

You can configure global default SNMP communities that are used if a VR does not exist on the router or if the community strings have not been configured for the VR.

Use the following configuration statements to configure global default SNMP communities:

```
shared sae configuration driver snmp {  
  read-only-community-string read-only-community-string;  
  read-write-community-string read-write-community-string;  
}
```

To configure global default SNMP communities:

1. From configuration mode, access the statements that configure default SNMP communities.

 user@host# edit shared sae configuration driver snmp
2. Configure the default SNMP community string used for read access to the router.

 [edit shared sae configuration driver snmp]
 user@host# set read-only-community-string *read-only-community-string*
3. Configure the default SNMP community string used for write access to the router.

```
[edit shared sae configuration driver snmp]
user@host# set read-write-community-string read-write-community-string
```

4. (Optional) Verify your configuration.

```
[edit shared sae configuration driver snmp]
user@host# show
read-only-community-string *****;
read-write-community-string *****;
```

Using the NIC Resolver in Environments That Have Third-Party Devices (SRC CLI)

If you are using the assigned IP subscriber method of logging in subscribers, and you are using the NIC to determine the subscriber's SAE, you need to configure a resolver on the NIC. The OnePopDynamicIp sample configuration data supports this scenario. The OnePopDynamicIp configuration supports one point of presence (POP) and provides no redundancy. The realm for this configuration accommodates the situation in which IP pools are configured locally on each virtual router object.

You can access the OnePopDynamicIp configuration in the SRC CLI.

- Related Topics**
- Configuration Tasks for Integrating Third-Party Network Devices (SRC CLI) on page 96
 - Overview of Integrating Network Devices into the SRC Network on page 91
 - Configuring the NIC (SRC CLI) on page 130

PART 4

Locating Subscriber Management Information

- Locating Subscriber Information with the NIC on page 111
- Configuring the NIC (SRC CLI) on page 127
- Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrfIp on page 149
- Configuring Applications to Communicate with an SAE on page 163
- Configuring SRC Applications to Communicate with an SAE (SRC CLI) on page 165
- Developing Applications That Use NIC on page 173
- NIC Resolution Process on page 181
- NIC Configuration Scenarios on page 187

CHAPTER 9

Locating Subscriber Information with the NIC

- Locating Subscriber Management Information on page 111
- Mapping Subscribers to a Managing SAE on page 113
- High Availability for NIC on page 114
- Planning a NIC Implementation on page 117
- NIC Configuration Scenarios on page 118
- NIC Agents Used in the NIC Configuration Scenarios on page 122
- Router Initialization Scripts with NIC Configuration Scenarios on page 124

Locating Subscriber Management Information

For services to be activated for a subscriber session, applications such as the SRC Volume-Tracking Application (SRC VTA), Dynamic Service Activator, Enterprise Manager Portal, or a residential portal need to locate the SAE that manages the subscriber. An application such as the Threat Mitigation Application Portal needs to locate the SAE that manages interfaces through which traffic destined for a specified IP address enters the network.

The NIC is the component that locates which SAE manages a subscriber or an interface. The NIC uses information that identifies the subscriber or the interface to identify the managing SAE. A NIC is similar to a Domain Name System (DNS) in that a NIC processes resolution requests. Rather than translating hostnames to IP addresses and vice versa, the NIC resolves an identifier for a subscriber or an interface to a reference for the managing SAE.

The components that participate in this resolution are a NIC host and a NIC proxy, also called a NIC locator for particular applications. A NIC host processes resolution requests. A NIC proxy requests data resolution for an application. A NIC proxy is so-named because it requests information on behalf of an application. A NIC proxy and a NIC host communicate with each other through Common Object Request Broker Architecture (CORBA); NIC manages the CORBA interactions for you.

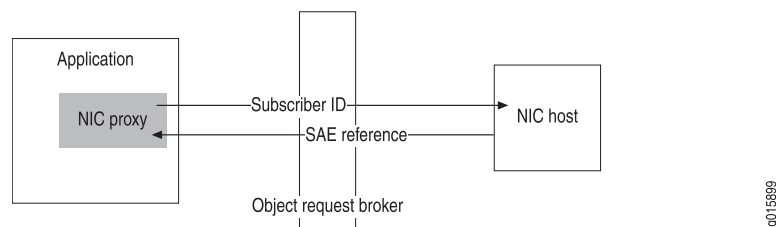
NIC can operate in a client/server mode or in a local host mode. In the client/server mode, a NIC host and NIC proxies can reside on different systems. In local host mode, a NIC host and NIC proxies reside in the same process on a machine.

NIC Client/Server Mode

In client/server mode, a NIC host is the server. A NIC proxy, which comprises libraries within an application that interacts with a NIC host, is the client.

Figure 10 on page 112 shows a NIC proxy running within an application and a NIC host running on a different machine. Both communicate through CORBA, with the NIC proxy providing an identifier for a subscriber and the NIC host returning a reference to the SAE that manages the subscriber.

Figure 10: Communication Between a NIC Proxy and a NIC Host in Client/Server Mode



NIC Local Host Mode

In local host mode, a Java application can include the libraries for a NIC host as well as NIC proxies. With this configuration, the NIC host and the NIC proxies communicate with each other within the same application. Because both components run within the same application, the application and the NIC host start and stop at the same time.

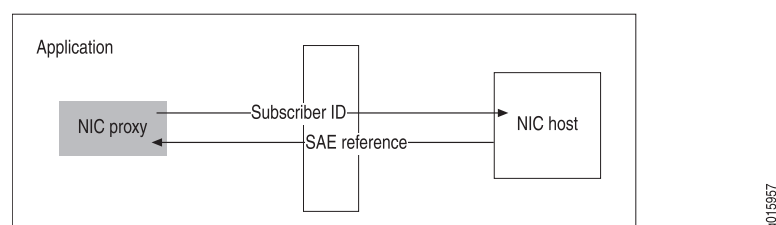
If an application uses a local NIC host, all NIC proxies for the application typically communicate with the local NIC host, but some of the NIC proxies can be configured to communicate with a NIC host that runs on another system.

When you use NIC in local host mode:

- You cannot use the C-Web interface to monitor or troubleshoot the local NIC host
- The NIC host runs all the resolvers and agents for the host on the local machine.
- Other NIC hosts cannot communicate with agents and resolvers that run in a local NIC host.

Figure 11 on page 112 shows a NIC proxy and a NIC host running within an application.

Figure 11: Communication Between a NIC Host and a NIC Proxy in Local Host Mode



- Related Topics**
- Mapping Subscribers to a Managing SAE on page 113
 - High Availability for NIC on page 114
 - Overview of NIC Proxy Configuration on page 163
 - NIC Configuration Scenarios on page 118
 - NIC Agents Used in the NIC Configuration Scenarios on page 122

Mapping Subscribers to a Managing SAE

A NIC collects information about the state of the network and can provide mapping from a specified type of network data, known as a *key*, to another type of network data, known as a *value*. Applications can use a NIC proxy to submit a key to a NIC host. The NIC host obtains a corresponding value from other components within NIC and returns it through the NIC proxy to the application. A typical use of a NIC is for a residential portal application to submit a subscriber's IP address and for the NIC to return the interoperable object reference (IOR) of the SAE managing that subscriber.

NIC Proxies and NIC Locators

Typically, an application supports one NIC proxy for each type of data request. A NIC proxy caches resolution results for a period of time so that it can resolve future requests without consulting the NIC host, thereby decreasing traffic between the NIC proxy and the NIC host. Applications that use NIC proxies communicate with the proxy to delete any invalid cache entries. Caching lets you optimize resolution performance for your network configuration and system resources.

You configure a NIC proxy when you configure that application. SRC applications such as the SRC VTA and Dynamic Service Activator contain NIC proxies. If you are writing an external application that will interact with a NIC, you must include NIC proxies in the application.

A NIC locator provides the same functionality as a NIC proxy; however, it runs as part of the NIC host. A NIC locator uses the NIC access interface module, a simple CORBA interface, to enable non-Java applications to interact with NIC. A NIC locator does not cache information.

For information about the NIC access interface module, see the API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.

For more information about NIC proxies and NIC locators, see “Overview of NIC Proxy Configuration” on page 163.

NIC Hosts

NIC hosts collect and store SRC information, and respond to requests from NIC proxies. The components in a NIC host that manage this process are:

- NIC agents—Collect data from SRC components, publish data, and make data available to NIC resolvers

- NIC resolvers—Process resolution requests

NIC Agents

NIC agents collect information about the state of the network from many data sources on the network. Table 7 on page 114 describes the types of agents supplied with NIC.

Table 7: Types of NIC Agents

Type of Agent	Type of Information the Agent Makes Available
Consolidator agent	Summary information received from other agents.
Directory agent	Specified directory entries and changes to directory entries.
Properties agent	Information from a specified list of property file. Typically, you do not configure properties agents.
SAE client agent	SAEs managing a subscriber at resolution time.
SAE plug-in agent	Subscriber information and interface information for SAE-managed subscribers and interfaces.
SSR client agent	Subscriber information from the Session State Registrar.
XML agent	Information from a specified XML document. Typically, you do not configure XML agents.

NIC Resolvers

NIC resolvers manage information to resolve requests by:

- Receiving and storing information about the state of the network from components within NIC and other NIC resolvers
- Requesting information from NIC agents and other NIC resolvers
- Receiving requests from the NIC proxies or other NIC resolvers
- Processing requests and sending responses to the requesters

- Related Topics**
- Locating Subscriber Management Information on page 111
 - Configuring a NIC Scenario (SRC CLI) on page 134

High Availability for NIC

You can configure high availability for NIC when you use client/server mode with the NIC host and the NIC proxies running on different machines. NIC supports several mechanisms to maintain high availability. We recommend that you use NIC replication to keep a NIC

configuration highly available. NIC replication uses groups of NIC hosts that share the same configuration for NIC resolutions to respond to resolution requests.

When you use NIC in local host mode, you do not need to configure redundancy for a NIC host, because the NIC host runs within the application.

High Availability in Existing NIC Configurations

If you have a previous NIC configuration, you may be using:

- NIC host redundancy, in which a set of NIC hosts provide redundancy

The SRC CLI does not support NIC host redundancy.

- Redundancy for SAE plug-in agents, in which a set of SAE plug-in agents provide redundancy

If you have an SAE plug-in agent that uses agent redundancy, enable state synchronization for the agent and use NIC replication. In SRC Release 1.0.0, configuration for SAE plug-in agent redundancy is discontinued.

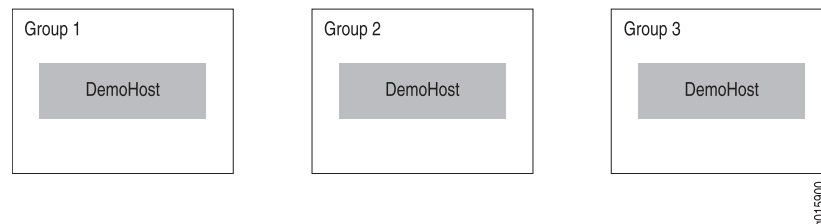
NIC Replication

NIC replication uses the concept of a group to identify a NIC host that has a particular configuration. A group contains one or more NIC hosts; each NIC host in a group is unique; for example, each NIC host could reside on a different system. A NIC proxy contacts specified groups that contain hosts with the same configuration to locate a managing SAE.

For example, a group might include the host DemoHost, but not two instances of DemoHost. Typically, each NIC host in a group is located in the same point of presence (POP). However, a machine can support only one NIC host. The SRC software stores groups in the directory in *ou=dynamicConfiguration*, *ou=Configuration*, *o=Management*, *o=umc*.

For example, Figure 12 on page 115 shows three NIC groups with each group containing a NIC host that has the same configuration.

Figure 12: NIC Groups



Groups let you:

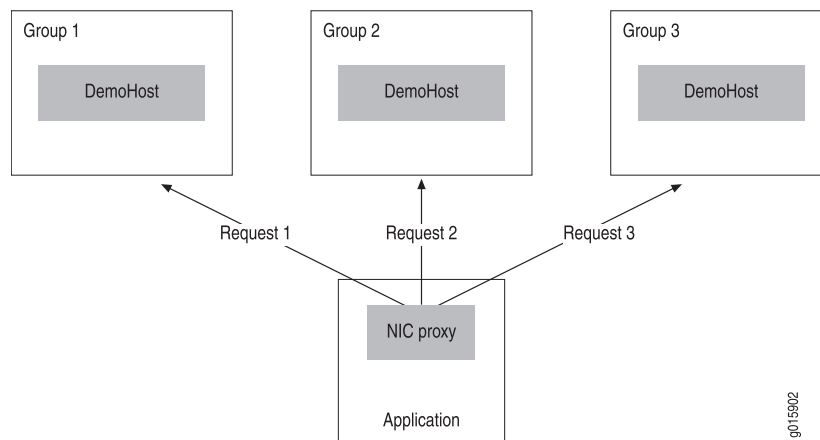
- Distribute network and processing load between two or more groups
- Provide failover protection if one group becomes unavailable

With NIC replication, a NIC proxy can contact multiple NIC hosts that are assigned to different groups. When a NIC proxy is configured to contact more than one group, the NIC configuration on a NIC host in each group should be equivalent—the NIC hosts should use the same configuration scenarios.

A NIC proxy selects a group by using the method specified in the configuration for the proxy; for example, the NIC proxy can randomly choose a group from a list. The NIC proxy then sends resolution requests to the corresponding host in that group. If a NIC proxy submits high numbers of resolution requests to the NIC host, you can configure the NIC proxy to randomly pick a NIC host or to pick a NIC host in a cyclic order to decrease the probability that one NIC host manages all the resolution requests.

Figure 13 on page 116 shows resolution requests sent by means of a round-robin selection.

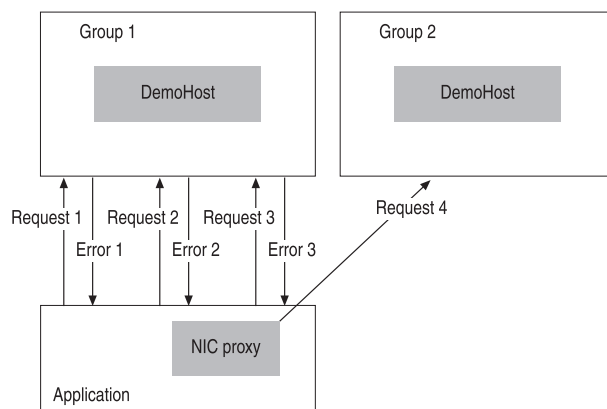
Figure 13: NIC Group Selection by Round-Robin



If the NIC host fails to respond to a specified number of resolution requests, the NIC proxy stops sending resolution requests to the unavailable NIC host and sends the resolution requests to another NIC host. The NIC proxy continues to poll the unavailable NIC host to determine its availability. When the NIC host becomes available, the NIC proxy can again send resolution requests to that host.

Figure 14 on page 117 shows a NIC proxy that sends a resolution request to Group 1, receives an error message, then sends two more resolution requests before sending a request to Group 2 rather than Group 1. When Group 1 is available again, the NIC proxy will send the request to Group 1.

Figure 14: NIC Resolution Request



g015903

You configure NIC replication for hosts, then configure NIC proxies to use replication.

Although you can distribute agents and resolvers among different hosts, as shown in the configuration for the NIC hosts OnePopBO and OnePopH1 in the sample data, we recommend that you use the DemoHost configuration, which centralizes the configuration for agents and resolvers.

- Related Topics**
- Router Initialization Scripts with NIC Configuration Scenarios on page 124
 - Planning a NIC Implementation on page 117
 - NIC Configuration Scenarios on page 118
 - NIC Agents Used in the NIC Configuration Scenarios on page 122

Planning a NIC Implementation

The SRC software provides standard NIC configuration scenarios that you can modify to meet the requirements for your environment. Which scenarios you choose depends on the applications you use.

If the resolution scenarios do not provide the type of resolution needed, we recommend that you consult Juniper Professional Services.

To plan your NIC implementation:

1. Review the NIC configuration scenarios, and select the scenario that best fits the requirements for your application. In most cases, one of the basic configuration scenarios provides the type of resolution needed.
See "NIC Configuration Scenarios" on page 118.
2. Determine the number of NIC proxies that you will need to access NIC hosts, and estimate the amount of traffic between the NIC proxies and the NIC hosts. If you expect heavy traffic between NIC proxies and NIC hosts, configure a number of NIC hosts to share the traffic load and processing.

3. Determine which NIC hosts to assign to a group to provide NIC replication; choose names for these groups.
4. If you have not done so already, determine which systems are to run NIC hosts.

- Related Topics**
- Testing a NIC Resolution (SRC CLI) on page 144
 - Router Initialization Scripts with NIC Configuration Scenarios on page 124
 - NIC Agents Used in the NIC Configuration Scenarios on page 122
 - Overview of the NIC Resolution Process on page 181

NIC Configuration Scenarios

Table 8 on page 118 lists the NIC configuration scenarios provided in the SRC software.

Table 8: NIC Configuration Scenarios

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
Basic Configuration Scenarios			
For JUNOS local configuration for PPP and DHCP subscribers. Sample use: DSL providers for residential customers.	OnePop	Subscriber IP address to the SAE IOR	Simplest configuration. IP pools configured locally on each virtual router (VR) with IP addresses from a static pool of IP addresses configured on the virtual router.

Table 8: NIC Configuration Scenarios (*continued*)

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
<p>For subscribers who have an accounting ID.</p> <p>Can be used for multiple subscribers who use the same accounting ID, in which case NIC returns all SAE IORs for mapped subscribers.</p> <p>Sample use:</p> <p>Support for the volume-tracking application.</p>	OnePopAcctId	Accounting ID of a subscriber to the SAE IOR and the IP address of a subscriber to accounting ID	<p>A subscriber's accounting ID can be specified at subscriber login from the SAE subscriber classification script. As a result, the accounting ID encapsulates other attributes of the subscriber session processed by the subscriber classification script. The OnePopAcctId configuration scenario can resolve the encapsulated attributes.</p> <p>For example, customers can assign a subscriber username (login id without domain name) to an accounting ID with the following subscriber classification.</p> <pre>[<ID>=accounting ID=<name>=<ip>=<-userName->]]</pre>
<p>For subscribers who have assigned IP addresses (assigned external to the SAE).</p> <p>Sample use:</p> <p>In a PacketCable Multimedia Specification (PCMM) environment when the SAE acts as both a policy server and application manager.</p>	OnePopDynamicIp	Subscriber IP address to the SAE IOR	
<p>For resolution of a subscriber login name to an SAE IOR, and of a subscriber IP address to a subscriber login name.</p> <p>Sample use:</p> <p>Support for tracking subscriber bandwidth usage or for using a billing model. You can use the SRC VTA with this scenario.</p>	OnePopLogin	Subscriber login name to the SAE IOR and subscriber IP address to login name	Uses two resolvers. Use a separate NIC proxy for each resolution.
For use with applications that need to support tracking a large number of subscribers.	OnePopLoginPull	Subscriber login name or a subscriber IP address to an SAE IOR	

Table 8: NIC Configuration Scenarios (*continued*)

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
<p>For subscribers who connect through a cable modem termination system (CMTS) device.</p> <p>Sample use:</p> <p>In a PCMM environment in which the policy server is separate from the application server. This scenario can be used when the configuration includes Juniper Policy Server or another policy server, and the SAE is an application manager.</p>	OnePopPcmm	Subscriber IP address to the SAE IOR	
<p>For use with applications that use the SAE programming interfaces and that identify subscribers by the primary username.</p> <p>Sample uses:</p> <ul style="list-style-type: none"> Aggregate services Dynamic service activator application 	OnePopPrimaryUser	Primary username of a subscriber to the SAE IOR	Similar to OnePopLogin
<p>For a router configuration in which VRs share IP pools.</p> <p>Sample use:</p> <ul style="list-style-type: none"> Services for enterprise subscribers. Support for two different proxies: Subscriber DN to the SAE IOR Subscriber IP address to the SAE IOR 	OnePopDnSharedIp	Subscriber distinguished name (DN) or subscriber IP address to the SAE IOR	Includes resolution available in OnPopSharedIp and adds resolution from a subscriber DN.
<p>For a router configuration in which pools can be shared among routers. Pools can be assigned by RADIUS or by a DHCP server.</p> <p>Sample use:</p> <p>Support for DHCP and PPP connections for residential subscribers.</p>	OnePopSharedIp	Subscriber IP address to the SAE IOR	

Table 8: NIC Configuration Scenarios (*continued*)

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
For scenarios in which subscribers have an assigned IP address and these IP addresses can be associated with interfaces on JUNOS routing platforms. Sample use: <ul style="list-style-type: none"> Threat Mitigation Application Portal 	OnePopStaticRouteIp	Assigned subscriber IP address to the SAE IOR	Static route information for routers resides in an XML document in the directory under the router object.
For scenarios in which subscribers have an assigned IP address. Sample use: <ul style="list-style-type: none"> Applications that use an SAE to manage a provider edge router, not directly manage end subscribers, and not support individual subscriber sessions for these subscribers. 	OnePopVrflp	Assigned subscriber IP address to the SAE IOR	Similar to OnePopStaticRouteIp. Used to support multiple VPNs with overlapping IP pools. Static route information for routers resides in an XML document in the directory under the router object.
For enterprise customers.	OnePopAllRealms	Subscriber IP address or subscriber DN to the SAE IOR	The scenario combines the OnePop and OnePopSharedIp scenarios and adds resolution from a subscriber DN.
Advanced Configuration Scenario			
For two POPs that share a back office. Sample use: Support for a deployment that has a back office that connects to NIC hosts at other sites.	MultiPop	Subscriber IP address to the SAE IOR	You can deploy this scenario in an environment that has a number of POPs; for example, a configuration in which there are two POPs with NIC proxy communication to a back office, which in turn communicates with the POP hosts. The POP hosts each support parallel hosts and agents and manage resolutions in the same way. You can add POPs by copying the configuration for one POP and modifying the configuration to suit your environment.

Related Topics • Configuration Statements for the NIC on page 127

- Router Initialization Scripts with NIC Configuration Scenarios on page 124
- NIC Agents Used in the NIC Configuration Scenarios on page 122

NIC Agents Used in the NIC Configuration Scenarios

When you configure a NIC configuration scenario, you use the basic configuration for each NIC agent in the scenario, but modify properties such as directory properties to make the agent configuration compatible with your SRC configuration. The NIC configuration scenario that you use determines which agents appear in your configuration.

Table 9 on page 122 lists all agents that are available in the various configuration scenarios.

Table 9: NIC Agents

Agent Name	Type of Agent	Type of Information
AcctIdIp	SAE plug-in	Mappings of accounting IDs of a subscribers to the SAE IOR and subscriber IP addresses to accounting ID(s).
DnVr	SAE plug-in	Mappings of enterprise access DNs to VRs.
Enterprise	Directory	List of enterprise names.
IpAcctId	SAE plug-in	Mappings of subscriber IP addresses to accounting IDs.
IpLoginName	SAE plug-in	Mappings of IP addresses to login names.
IpSaeld	SAE client	Mappings of IP addresses to SAEs managing a subscriber. Uses the SAE remote interface to determine which SAEs are managing a subscriber at resolution time.
IpVr	SAE plug-in	Mappings of IP addresses to VRs.
LoginNameVr	SAE plug-in	Mappings of login names to VRs.
LoginSaeld	SAE client	Mappings of login names to SAEs . Uses the SAE remote interface to determine which SAEs are managing a subscriber at resolution time.
PoolInterface	Directory	Mappings of IP pools to an interface. Note: Reads a JUNOS routing table and extracts the VR name to perform the mapping.
PoolVr	Directory	Mappings of IP pools to VRs.
UserNameVr	SAE plug-in	Mappings of subscriber IP addresses to accounting IDs.
VrSaeld	Directory	Reads information about virtual routers and the mappings between virtual routers and SAEs.

Table 10 on page 123 shows the types of agents that each configuration scenario uses.

Table 10: Agents in Configuration Scenarios

NIC Configuration Scenario	Directory Agents	SAE Plug-In Agents	SAE Client Agents	SSR Client Agents
OnePop	PoolVr, VrSaeld			
OnePopAcctId	PoolVr, VrSaeld	AcctIdIp, IpAcctId		
OnePopDnSharedIp	PoolVr, VrSaeld, Enterprise	DnVr		
OnePopDynamicIp	PoolVr, VrSaeld			
OnePopLogin	Pool, VrSaeld	IpLoginName, LoginNameVr		
OnePopLoginPull			IpSaeld, LoginSaeld	
OnePopPcmm	PoolVr, VrSaeld			
OnePopSharedIp	PoolVr, VrSaeld	IpVr		
MultiPop	PoolVr, VrSaeld, site-specific versions of PoolVr and VrSaeld	IpVr		
OnePopAllRealms	PoolVr, VrSaeld, Enterprise	IpVr		
OnePopPrimaryUser	VrSaeld	UserNameVr		
OnePopStaticRouteIp	VrSaeld, PoolInterface			
OnePopVrflp	VrSaeld, PoolInterface			

Related Topics

- Mapping Subscribers to a Managing SAE on page 113
- Router Initialization Scripts with NIC Configuration Scenarios on page 124
- Configuring a NIC Scenario (SRC CLI) on page 134
- NIC Configuration Scenarios on page 118

Router Initialization Scripts with NIC Configuration Scenarios

The NIC resolutions map VRs to SAEs. For these resolutions, use a router initialization script that associates each VR with the SAE that manages it. Which router initialization script you use depends on whether the SAE obtains IP pools from JUNOS VRs:

- **poolPublisher** router initialization script—Use when the SAE obtains local IP pools locally from JUNOS VRs.
- **iorPublisher** router initialization script—Use when the router is one of the following:
 - JUNOS routers that do not supply IP addresses from local pools
 - JUNOS routing platforms
 - CMTS devices

These devices do not supply IP addresses from local pools in your network.

Table 11 on page 124 lists which type of initialization script should be used with the various NIC configuration scenarios. The OnePopLoginPull scenario does not require an initialization script.

Table 11: Type of Router Initialization Script to Use for NIC Configuration Scenarios

poolPublisher	iorPublisher	poolPublisher or iorPublisher
One Pop	OnePopDnSharedIp	OnePopAcctId
	OneLoginPull	OnePopAllReams
	OnePopPcmm	OnePopDynamicIp
	OnePopPrimaryUser	OnePopLogin
	OnePopSharedIp	MultiPop
	OnePopStaticRouteIp	
	OnePopVrflp	



NOTE: If you modify information about IP pools on a VR after the COPS connection is established, the SAE does not automatically register the changes, and you must update the directory.

For more information about router initialization scripts for JUNOS routers, including how to update the directory, see “Configuring the SAE to Manage JUNOS Routers (SRC CLI)” on page 50.

For more information about router initialization scripts for JUNOS routing platforms, see “Configuring the SAE to Manage JUNOS Routing Platforms (SRC CLI)” on page 69.

- Related Topics**
- High Availability for NIC on page 114
 - Planning a NIC Implementation on page 117
 - NIC Configuration Scenarios on page 118
 - NIC Agents Used in the NIC Configuration Scenarios on page 122

CHAPTER 10

Configuring the NIC (SRC CLI)

- Configuration Statements for the NIC on page 127
- Before You Configure the NIC on page 129
- Configuring the NIC (SRC CLI) on page 130
- Reviewing and Changing Operating Properties for the NIC (SRC CLI) on page 131
- Configuring NIC Replication (SRC CLI) on page 133
- Configuring a NIC Scenario (SRC CLI) on page 134
- Configuring Advanced NIC Features on page 143
- Verifying Configuration for the NIC (SRC CLI) on page 143
- Starting the NIC (SRC CLI) on page 144
- Testing a NIC Resolution (SRC CLI) on page 144
- Stopping a NIC Host on a C Series Controller (SRC CLI) on page 145
- Restarting the NIC (SRC CLI) on page 145
- Restarting a NIC Agent (SRC CLI) on page 146
- Restarting a NIC Resolver (SRC CLI) on page 146
- Changing NIC Configurations (SRC CLI) on page 147

Configuration Statements for the NIC

The SRC CLI provides the following groups of configuration statements for the NIC:

- Configuration statements for NIC operating properties
- Configuration statements for NIC scenarios
- Configuration statements for NIC logging



NOTE: We recommend that you change only those statements visible at the basic editing level. Contact Juniper Professional Services or Juniper Customer Support before you change any of the NIC statements and options not visible at the basic editing level.

Configuration Statements for NIC Operating Properties

Use the following configuration statements to configure the NIC operating properties at the **[edit]** hierarchy level. These statements are visible at the CLI basic editing level.

```
slot number nic {
  base-dn base-dn;
  java-garbage-collection-options java-garbage-collection-options;
  java-heap-size java-heap-size;
  scenario-name scenario-name;
  snmp-agent;
  hostname hostname;
  runtime-group runtime-group;
}
slot number nic initial {
  static-dn static-dn;
  dynamic-dn dynamic-dn;
}
slot number nic initial directory-connection {
  url url;
  backup-urls [ backup-urls...];
  principal principal;
  credentials credentials;
  protocol (ldaps);
  timeout timeout;
  check-interval check-interval;
  blacklist;
  snmp-agent;
}
slot number nic initial directory-eventing {
  eventing;
  signature-dn signature-dn;
  polling-interval polling-interval;
  event-base-dn event-base-dn;
  dispatcher-pool-size dispatcher-pool-size;
}
```

Configuration Statements for NIC Scenarios

Use the following configuration statements to configure the NIC at the **[edit]** hierarchy level. These statements are visible at the CLI basic editing level.

Which agents you configure depends on the NIC configuration scenario that you use.



NOTE: Although the CLI provides configuration statements for SSR Client agents, you typically do not need to change the basic configuration for these agents. Changes can be made at the expert editing level.

The CLI also provides configuration statements for consolidator agents, properties agents, and XML agents. At this time, none of the NIC configuration scenarios uses these agents. The following list does not include the configuration statements for these agents.

```

shared nic scenario name
shared nic scenario name agents name
shared nic scenario name agents name configuration directory {
  search-base search-base ;
  search-filter search-filter ;
  search-scope (0 | 1 | 2);
  server-url server-url ;
  directory-backup--urls directory-backup-urls ;
  principal principal ;
  credentials credentials ;
}
shared nic scenario name agents name configuration sae-client {
  principal principal;
  credentials credentials;
  subscriber-id (user-ip-address | dn | login-name | interface-name | primary-user-name);
  search-base search-base;
  search-filter search-filter;
  search-scope (object | one-level | sub-tree);
  server-url server-url;
  directory-backup-urls directory-backup-urls ;
}
shared nic scenario name agents agent configuration sae-plug-in {
  event-filter event-filter ;
  number-of-events number-of-events ;
}

```

Configuration Statements for NIC Logging

Use the following configuration statements to configure logging for the NIC at the **[edit]** hierarchy level.

```

shared nic scenario name hosts name configuration logger name syslog {
  filter filter ;
  host host ;
  facility facility ;
  format format ;
}
shared nic scenario name hosts name configuration logger name file {
  filter filter ;
  filename filename;
  rollover-filename rollover-filename ;
  maximum-file-size maximum-file-size ;
}

```

- Related Topics**
- Before You Configure the NIC on page 129
 - Configuring the NIC (SRC CLI) on page 130
 - For detailed information about each configuration statement, see the *SRC PE CLI Command Reference*.

Before You Configure the NIC

When you use NIC in a client/server configuration, you configure the NIC scenario before you configure the NIC proxies.

Before you configure NIC hosts from the CLI:

- Plan your NIC implementation:
- Choose the NIC configuration scenario to use.

The default scenario is OnePop.

For information about NIC configuration scenarios and NIC agents, see “Locating Subscriber Management Information” on page 111.

- Ensure that the appropriate type of router initialization script is configured for the router or network device.

See “Locating Subscriber Management Information” on page 111.

Set the editing level for the configuration application you are using, the SRC CLI or the C-Web interface to basic. This ensures that only the statements that you need to configure are visible.

To set the editing level for the C-Web interface to basic:

- Click **Preferences>Level Basic**.

Related Topics

- Configuring the NIC (SRC CLI) on page 130
- Configuring the NIC (C-Web Interface)
- Starting the NIC (SRC CLI) on page 144
- NIC Agents Used in the NIC Configuration Scenarios on page 122
- Router Initialization Scripts with NIC Configuration Scenarios on page 124
- Verifying Configuration for the NIC (SRC CLI) on page 143

Configuring the NIC (SRC CLI)

Before you configure the NIC, complete the prerequisite tasks.

See “Before You Configure the NIC” on page 129.

To configure the NIC:

1. Configure NIC operating properties.
See “Reviewing and Changing Operating Properties for the NIC (SRC CLI)” on page 131.
2. Configure NIC replication.
See “Reviewing and Changing Operating Properties for the NIC (SRC CLI)” on page 131.
3. (Optional) If you plan to use a configuration scenario other than OnePop (the default), delete any data for the OnePop scenario and configure the scenario name to specify the configuration scenario.
See “Changing NIC Configurations (SRC CLI)” on page 147.
4. Configure a NIC scenario.

See “Configuring a NIC Scenario (SRC CLI)” on page 134.

5. Verify the NIC configuration.

See “Verifying Configuration for the NIC (SRC CLI)” on page 143.

6. Start the NIC component.

See “Starting the NIC (SRC CLI)” on page 144.

- Related Topics**
- Testing a NIC Resolution (SRC CLI) on page 144
 - Configuration Statements for the NIC on page 127

Reviewing and Changing Operating Properties for the NIC (SRC CLI)

Before you configure a NIC configuration scenario, review the default operating properties and change values as needed. Operating properties are configured for a slot.

The following topics provide procedures for reviewing and changing operating properties for NIC with the SRC CLI:

1. Reviewing the Default NIC Operating Properties on page 131
2. Changing NIC Operating Properties on page 132

Reviewing the Default NIC Operating Properties

To review the default NIC operating properties:

1. From configuration mode, access the configuration statement that specifies the configuration for the NIC on a slot.

```
[edit]
user@host# edit slot number nic
```

For example:

```
[edit]
user@host# edit slot 0 nic
```

2. Run the **show** command.

```
[edit slot 0 nic]
user@host# show
base-dn o=umc;
java-runtime-environment ../jre/bin/java;
java-heap-size 128m;
snmp-agent;
hostname DemoHost;
initial {
    dynamic-dn "ou=dynamicConfiguration, ou=Configuration,
o=Management,<base>";
    directory-connection {
        url ldap://127.0.0.1:389/;
        backup-urls ;
        principal cn=nic,ou=Components,o=Operators,<base>;
```

```
        credentials *****;  
        timeout 10;  
        check-interval 60;  
    }  
    directory-eventing {  
        eventing;  
        signature-dn <base>;  
        polling-interval 15;  
        event-base-dn <base>;  
        dispatcher-pool-size 1;  
    }  
    static-dn "l=OnePop,l=NIC, ou=staticConfiguration, ou=Configuration,  
o=Management,<base>";  
}
```

Changing NIC Operating Properties

In most cases you can use the default NIC operating properties. Change the default properties if needed for your environment.

To change NIC operating properties:

1. From configuration mode, access the configuration statement that specifies the configuration for the NIC on a slot.

```
[edit]  
user@host# edit slot number nic
```

For example:

```
[edit]  
user@host# edit slot 0 nic
```

2. (Optional) If you store data in the directory in a location other than the default, *o=umc*, change this value.

```
[edit slot 0 nic]  
user@host# set base-dn base-dn
```

3. (Optional) Configure the garbage collection functionality of the Java Virtual Machine.

```
[edit slot 0 nic]  
user@host# set java-garbage-collection-options java-garbage-collection-options
```

4. (Optional) If you determine that additional memory is needed, change the maximum memory size available to the (Java Runtime Environment) JRE.

```
[edit slot 0 nic]  
user@host# set java-heap-size java-heap-size
```

By default, the JRE can allocate 128 MB. Set to a value lower than the available physical memory to avoid low performance because of disk swapping.

If you use an SAE plug-in agent, we recommend that you increase the JVM max heap to a value in the range 400–500 MB.

If you need help to determine the amount of memory needed, contact Juniper Networks Customer Services and Support.

5. (Optional) Specify the name of the NIC scenario that you want to configure. The default scenario is OnePop.

```
[edit slot 0 nic]
user@host# set scenario-name scenario-name
```

6. (Optional) Enable viewing of SNMP counters through an SNMP browser.

```
[edit slot 0 nic]
user@host# set snmp-agent
```

7. (Optional) Change the name of the NIC host. Use the default name of the NIC host configured for a NIC scenario. In most cases, the NIC host name is DemoHost.

```
[edit slot 0 nic]
user@host# set hostname hostname
```

8. (Optional) Change the initial properties.

See Configuring Basic Local Properties.

- Related Topics**
- Reviewing and Changing Operating Properties for NIC (C-Web Interface)
 - Configuring the NIC (SRC CLI) on page 130
 - Configuration Statements for the NIC on page 127
 - Changing NIC Configurations (SRC CLI) on page 147
 - Verifying Configuration for the NIC (SRC CLI) on page 143

Configuring NIC Replication (SRC CLI)

You configure NIC replication to keep the NIC configuration highly available.

Before you configure NIC replication:

- Make sure that you understand how NIC groups are used.
See “Locating Subscriber Management Information” on page 111.
- Identify which NIC hosts are to provide redundancy for each other.
- Select a name for a group for each of these hosts.

To configure NIC replication:

1. From configuration mode, access the configuration statement that specifies the configuration for the agent.

```
[edit]
user@host# slot number nic
```

For example:

```
[edit]
```

```
user@host# slot 0 nic
```

2. Configure the runtime group for the NIC host.

```
[edit slot 0 nic]  
user@host# runtime-group runtime-group
```

For example:

```
[edit slot 0 nic]  
user@host# runtime-group group1
```

Related Topics

- Configuring the NIC (SRC CLI) on page 130

Configuring a NIC Scenario (SRC CLI)

The following topics provide procedures for configuring a NIC scenario with the SRC CLI:

- Defining the NIC Configuration to Use on page 134
- Configuring Directory Agents on page 137
- Configuring SAE Client Agents on page 139
- Configuring SAE Plug-In Agents on page 140
- Configuring the SAE to Communicate with SAE Plug-In Agents When You Use NIC Replication on page 141

Defining the NIC Configuration to Use

The OnePop configuration scenario is the default configuration for NIC. If you want to use another configuration scenario, you first clear data for the configuration scenario and change the scenario name that identifies the scenario, see “Changing NIC Configurations (SRC CLI)” on page 147.

When you select a NIC configuration scenario, the software adds the default configuration for most properties. You can modify the NIC properties, including those for agents.



CAUTION: We recommend that you change only those statements visible at the basic editing level. Contact Juniper Professional Services or Juniper Customer Support before you change any of the NIC statements not visible at the basic editing level.

To specify a NIC configuration scenario for NIC to use:

1. Make sure that the NIC component is running.

```
user@host> show component  
Installed Components  
Name          Version                               Status  
...  
nic           Release: 7.0 Build: GATEWAY.A.7.0.0.0168  running  
...
```

- ```
[edit]
user@host# edit shared nic scenario name
```

```
[edit]
user@host# edit shared nic scenario OnePopLogin
```

- ```
[edit shared nic scenario OnePopLogin]
user@host# show

hosts {
    DemoHost {
        configuration {
            hosted-resolvers "/realms/login/A1, /realms/login/B1,
/realms/login/C1, /realms/login/D1, /realms/ip/A1, /realms/ip/B1,
/realms/ip/C1";
            hosted-agents "/agents/LoginNameVr, /agents/VrSaeId,
/agents/IpLoginName,
/agents/PoolVr";
        }
    }
    OnePopB0 {
        configuration {
            hosted-resolvers "/realms/login/A1, /realms/login/C1, /realms/ip/A1,
/realms/ip/C1";
            hosted-agents /agents/VrSaeId;
        }
    }
    OnePopH1 {
        configuration {
            hosted-resolvers "/realms/login/B1, /realms/login/D1, /realms/ip/B1";

            hosted-agents "/agents/LoginNameVr, /agents/IpLoginName,
/agents/PoolVr";
        }
    }
}
agents {
    VrSaeId {
        configuration {
            directory {
                search-base o=Network,<base>;
                search-filter (objectclass=umcVirtualRouter);
                search-scope 2;
                server-url ldap://127.0.0.1:389/;
                backup-servers-url ;
                principal cn=nic,ou=Components,o=Operators,<base>;
                , ' ' ' ' ' ' ' ' ' ' 'credentials *****';
            }
        }
    }
    LoginNameVr {
        configuration {
```

```

        sae-plugin {
            event-filter "(&(! (PA_USER_TYPE=INTF)) (! (PA_LOGIN_NAME=[None])))";

            number-of-events-sent-in-a-synchronization-call 50;
        }
    }
    IpLoginName {
        configuration {
            sae-plugin {
                number-of-events-sent-in-a-synchronization-call 50;
            }
        }
    }
    PoolVr {
        configuration {
            directory {
                search-base o=Network,<base>;
                search-filter (objectclass=umcVirtualRouter);
                search-scope 2;
                server-url ldap://127.0.0.1:389/;
                backup-servers-url ;
                ' ' ' ' ' ' ' ' ' ' 'principal cn=nic,ou=Components,o=Operators,<base>;
                ' ' ' ' ' ' ' ' ' ' 'credentials *****;
            }
        }
    }
}

```

4. (Optional) Update logging configuration.

See Overview of Logging for SRC Components.

By default, NIC has the following logging enabled for a NIC host:

```

logger file-1 {
    file {
        filter !ConfigMgr,!DES,/debug-;
        filename var/log/nicdebug.log;
        rollover-filename var/log/nicdebug.alt;
        maximum-file-size 10000000;
    }
}
logger file-2 {
    file {
        filter /info-;
        filename var/log/nicinfo.log;
    }
}
logger file-3 {
    file {
        filter /error-;
        filename var/log/nicerror.log;
    }
}

```

5. For each agent that the NIC configuration scenario includes, if needed update NIC agent configuration to define properties specific to your environment, such as directory properties.

Each type of agent has different configuration properties. The output from the **show** command identifies the type of agent under the **agents** hierarchy. For example:

```
VrSaeId {
  configuration {
    directory {

LoginNameVr {
  configuration {
    sae-plugin {
```

Configuring Directory Agents

Use the following configuration statements to configure NIC directory agents:

```
shared nic scenario name agents agent configuration directory {
  search-base search-base ;
  search-filter search-filter ;
  search-scope (0 | 1 | 2);
  server-url server-url ;
  backup-servers-url backup-servers-url ;
  principal principal ;
  credentials credentials ;
}
```

To configure a directory agent:

1. From configuration mode, access the statement that specifies the configuration for the agent.

```
[edit]
user@host# edit shared nic scenario name agents agent configuration directory
```

For example:

```
[edit]
user@host# edit shared nic scenario OnePopLogin agents VrSaeId configuration
directory
```

2. Review the default configuration for the agent. For example:

```
[edit shared nic scenario OnePopLogin agents VrSaeId configuration
directory]
user@host# show
search-base o=Network,<base>;
search-filter (objectclass=umcVirtualRouter);
search-scope 2;
server-url ldap://127.0.0.1:389/;
directory-backup-urls ;
principal cn=nic,ou=Components,o=Operators,<base>;
credentials *****;
```

3. (Optional) Change the distinguished name (DN) of the location in the directory from which the agent should read information.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set search-base search-base
```

For example:

```
[edit shared nic scenario OnePop agents PoolVr configuration directory]
user@host# set search-base o=myNetwork,<base>
```

You can use <base> in the DN to refer to the globally configured base DN.

4. (Optional) Change the directory search filter that the agent should use.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set search-filter search-filter
```

For example:

```
[edit shared nic scenario OnePop agents PoolVr configuration directory]
user@host# set search-filter objectclass=umcVirtualRouter
```

5. (Optional) Change the location in the directory relative to the base DN from which the NIC agent can retrieve information.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set search-scope (0 | 1 | 2)
```

where:

- 0—Entry specified in the **search-base** statement
 - 1—Entry specified in the **search-base** statement and objects that are subordinate by one level
 - 2—Subtree of entry specified in the **search-base** statement
6. For an installation on a Solaris platform, specify the location of the directory in URL string format.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set server-url ldap://host:portNumber
```

For example, to specify the directory on a C Series Controller:

```
[edit shared nic scenario OnePop agents PoolVr configuration directory]
user@host# set server-url ldap://127.0.0.1:389/
```

7. List the URLs of redundant directories. Separate URLs with semicolons.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set directory-backup-urls backup-servers-urls
```

8. Specify the DN that contains the username that the directory server uses to authenticate the NIC agent.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set principal principal
```

For example:

```
[edit shared nic scenario OnePop agents PoolVr configuration directory]
user@host# set principal cn=nic,ou=Components,o=Operators,<base>
```

9. Specify the password that the directory server uses to authenticate the NIC agent.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set credentials credentials
```

10. Restart the NIC agent.

```
user@host>request nic restart agent name name
```

Configuring SAE Client Agents

Use the following configuration statements to configure NIC SAE client agents:

```
shared nic scenario nameagents nameconfiguration sae-client {
  principal principal;
  credentials credentials;
  subscriber-id (user-ip-address | dn | login-name | interface-name | primary-user-name);
  search-base search-base;
  search-filter search-filter;
  search-scope (object | one-level | sub-tree);
  server-url server-url;
  directory-backup-urlsdirectory-backup-urls ;
}
```

To configure an SAE client agent:

1. From configuration mode, access the statement that specifies the configuration for the agent.

```
[edit]
user@host# edit shared nic scenario name agents agent configuration sae-client
```

For example:

```
[edit]
user@host# edit shared nic scenario OnePopLoginPull agents IpSaeld configuration
sae-client
```

2. Review the default configuration for the agent. For example:

```
[edit shared nic scenario OnePopLoginPull agents IpSaeId configuration sae-client]
user@host# show
principal cn=umcadmin,<base>;
credentials *****;
subscriber-id user-ip-address;
search-base ou=sspadmurls,o=Servers;;
search-filter (objectclass=corbaObjectReference);
search-scope sub-tree;
server-url ldap://127.0.0.1:389/; directory-backup-urls "";
```

3. (Optional) Change the authentication DN.

For example:

```
[edit shared nic scenario OnePopLoginPull agents IpSaeld configuration sae-client
]
user@host# set principal cn=umcadmin, <base>
```

4. (Optional) Change the password that the NIC uses to access the directory. For example:

```
[edit edit shared nic scenario OnePopLoginPull agents IpSaeld configuration sae-client
]
user@host# set credentials —
```

5. Specify the part of the directory that you want the network publisher to search.

```
[edit edit shared nic scenario OnePopLoginPull agents IpSaeld configuration sae-client
]
user@host# set search-base search-base
```

6. (Optional) Change the URL that identifies the primary Juniper Networks database to which the NIC agent connects.

```
[edit edit shared nic scenario OnePopLoginPull agents IpSaeld configuration sae-client
]
user@host# set server-url server-url
```

7. Specify the type of subscriber ID that the agent uses to identify the subscriber. The type can be **user-ip-address**, **dn**, **login-name**, or **interface-name**. For example, to specify an IP address:

```
[edit edit shared nic scenario OnePopLoginPull agents IpSaeld configuration sae-client
]
user@host# set subscriber-id use-ip-address
```

Configuring SAE Plug-In Agents

By default, the CORBA naming server on a C Series Controller uses port 2809. The NIC host is configured to communicate with this naming server; you do not need to change JacORB properties.

Use the following configuration statements to configure NIC SAE plug-in agents:

```
shared nic scenario name agents agent configuration sae-plugin-in{
  event-filter event-filter ;
  number-of-events number-of-events ;
}
```

If you plan to change the event filter for the agent, make sure that you are familiar with:

- Plug-in attributes and values

See Types of Tracking Plug-Ins .

- Filter syntax

See the documentation for the SAE CORBA Remote API in the SAE Core API documentation on the Juniper Networks Web site at:

<http://www.juniper.net/techpubs/software/management/src/api-index.html>

To configure an SAE plug-in agent:

1. From configuration mode, access the statement that specifies the configuration for the agent.

```
[edit]
user@host# edit shared nic scenario name agents agent configuration sae-plug-in
```

For example:

```
[edit]
user@host# edit shared nic scenario OnePopLogin agents LoginNameVr configuration
sae plug-in
```

2. Review the default configuration for the agent. For example:

```
[edit shared nic scenario OnePopLogin agents LoginNameVr configuration sae-plug-in]
user@host# show
event-filter "&(! (PA_USER_TYPE=INTF)) (! (PA_LOGIN_NAME=[None]))";
number-of-events-sent-in-a-synchronization-call 50;
```

3. (Optional) Change an LDAP filter that change the events that the agent collects.

```
[edit shared nic scenario name agents agent configuration sae-plug-in]
user@host# set event-filter event-filter
```

Typically, you do not need to change this value. If you do want to filter other events, use the format ***pluginAttribute=attributeValue*** format for event filters, where:

- ***pluginAttribute*** —Plug-in attribute name
- ***attributeValue*** —Value of filter

For example:

```
[edit shared nic scenario name agents agent configuration sae-plug-in]
user@host# set event-filter PA_USER_TYPE=INTF
```

4. Specify the number of events that the SAE sends to the agent at one time during state synchronization.

```
[edit shared nic scenario name agents agent configuration sae-plug-in]
user@host# set number-of-events number-of-events
```

For example:

```
[edit shared nic scenario OnePopLogin agents LoginNameVr configuration sae plug-in]
user@host# set number-of-events 50
```

Configuring the SAE to Communicate with SAE Plug-In Agents When You Use NIC Replication

For each NIC host that uses SAE plug-in agents, configure a corresponding external plug-in for the SAE. By default, the SAE plug-in agents share events with the single SAE plug-in. You must also configure the SAE to communicate with the SAE plug-in agent in each NIC host that you use in the NIC replication.

For information about configuring an external plug-in for the SAE, see *Configuring the SAE for External Plug-Ins (SRC CLI)*.

To configure an external plug-in:

1. From configuration mode, access the statement that specifies the configuration for an external plug-in for the SAE that communicates with the agent, and assign the plug-in a unique name.

```
[edit]
user@host# shared sae configuration plug-ins name name
```

2. Configure CORBA object reference for the plug-in.

```
[shared sae configuration plug-ins name name external]
user@host# corba-object-reference corba-object-reference
```

For the CORBA object reference, use the following syntax:

host : port-number /NameService# pluginName

where:

- ***host*** —IP address or name of the machine on which you installed the NIC host that supports the agent

For local host, use the IP address 127.0.0.1.

- ***port-number*** —Port on which the name server runs

The default port number is 2809.

- ***pluginName*** —Name under which the agent is registered in the naming service

Use the format ***nicxae_ groupname /saePort*** where ***groupname*** is the name of the replication group. (When replication is not used, the format is ***nicxae/saePort***.)

For example:

```
[shared sae configuration plug-ins name name external]
user@host# set corba-object-reference
corbaname::127.0.0.1:2809/NameService#nicxae/saePort
```

3. Configure attributes that are sent to the external plug-in for a NIC host. Because the SAE plug-in agents share the event by default, you configure only one for a NIC host.

```
[shared sae configuration plug-ins name name external]
user@host# set attr
[( router-name | user-dn | session-id | user-type | user-ip-address | login-name)]
```

Specify the plug-in options that the agent uses. You must specify the options ***session-id*** and ***router-name***, and other options that you specified for the agent's network data types and the agent's event filter. Do not specify attributes options of the PAT_OPAQUE attribute type, such as the option ***dhcp-packet***.



NOTE: Do not include attributes that are not needed.

4. Reference the NIC as a subscriber tracking plug-in.

```
[edit shared sae group name configuration plugins event-publishers]
user@host# set subscriber-tracking pool-name
```

For example, for a pool named nic:

```
[edit shared sae group name configuration plugins event-publishers]
user@host# set subscriber-tracking nic
```

- Related Topics**
- Configuring a NIC Scenario (C-Web Interface)
 - Configuring the NIC (SRC CLI) on page 130
 - Verifying Configuration for the NIC (SRC CLI) on page 143
 - Configuration Statements for the NIC on page 127
 - Overview of NIC Configuration Scenarios on page 187

Configuring Advanced NIC Features

If you want to configure NIC features not available at the basic editing level, set the editing level to advanced or expert and use the CLI Help to obtain information about statement options.

- Related Topics**
- Configuring the NIC (SRC CLI) on page 130
 - Configuring a NIC Scenario (SRC CLI) on page 134
 - Configuring NIC to Store Log Messages in a File (C-Web Interface)

Verifying Configuration for the NIC (SRC CLI)

Purpose After you complete the NIC configuration, verify the local NIC configuration and the NIC configuration scenario information.

Action To verify NIC configuration:

1. In configuration mode, run the **show** command at the **[edit slot 0 nic]** hierarchy level.

```
[edit slot 0 nic]
user@host# show
```
2. In configuration mode, run the **show** command at the **[edit shared nic scenario *name*]** hierarchy level.

For example:

```
[edit shared nic scenario OnePop]
user@host# show
```

- Related Topics**
- Starting the NIC (SRC CLI) on page 144

- Configuring the NIC (SRC CLI) on page 130
- Testing a NIC Resolution (SRC CLI) on page 144
- Changing NIC Configurations (SRC CLI) on page 147

Starting the NIC (SRC CLI)

Start the NIC component before you configure it. When you enable NIC for the first time, it creates the default operating properties for the component.

To start NIC:

- From operational mode, enable the NIC.

```
user@host> enable component nic
```

Starting NICHOST: may take a few minutes...

Related Topics

- Starting the NIC (C-Web Interface)
- Configuring the NIC (SRC CLI) on page 130
- Reviewing and Changing Operating Properties for the NIC (SRC CLI) on page 131
- Restarting the NIC (SRC CLI) on page 145
- Stopping a NIC Host on a C Series Controller (SRC CLI) on page 145

Testing a NIC Resolution (SRC CLI)

To test a NIC resolution:

- Run the **test nic resolve** command.

```
user@host> test nic resolve <locator locator> <key key>
```

where:

- **locator** —Name of locator that requests information on behalf of an application
- **key** —Value to be resolved. This value must be of the same NIC data type configured in the NIC locator.

For example:

```
user@host> test nic resolve locator /nicLocators/ip key 10.10.10.10
```

Example: Testing a NIC Resolution

The following example shows a successful resolution for an IP key that has the value 192.168.8.2:

```
user@host> test nic resolve locator /nicLocators/ip key 192.168.8.2
IOR:
0000000000000354944C3A738D67742E6A75E697065722E6E65742F7361652F5365727669636541637469766174696F6E456E67696E653A312E3000
0000000000010000000000006800010200000000F3137322E32382E3233302E313230000022610000000001073726320382F736165504F412F53
```


The following example shows an unsuccessful resolution for an IP key that has the value 192.168.8.2:

Related Topics

- [Configuring NIC Test Data \(SRC CLI\) on page 171](#)
- [Testing a NIC Resolution \(C-Web Interface\)](#)
- [Stopping a NIC Host on a C Series Controller \(SRC CLI\) on page 145](#)

To stop a NIC host:

- ```
user@host> disable component nic
```

```
user@host> request restart nic
```

You can also restart the NIC at the slot level.

- 145

- Restarting the NIC (C-Web Interface)

---

## Restarting a NIC Agent (SRC CLI)

You can restart a NIC agent to have the agent read all data in the directory again. Restart a NIC agent if the agent is not synchronized with the directory, or if you switch from one directory to another.

To restart a NIC agent:

- From operational mode, restart the agent.  
`user@host>request nic restart agent name name`

You can restart all NIC agents by omitting an agent name for the **request nic restart agent** command.

You can also restart a NIC agent at the slot level.

- Related Topics**
- Stopping a NIC Host on a C Series Controller (SRC CLI) on page 145
  - Restarting the NIC (SRC CLI) on page 145
  - Restarting a NIC Resolver (SRC CLI) on page 146
  - Changing NIC Configurations (SRC CLI) on page 147

---

## Restarting a NIC Resolver (SRC CLI)

In rare instances, such as when you are troubleshooting a NIC configuration, you may want to restart a NIC resolver.

To restart a NIC resolver:

- From operational mode, restart a resolver.  
`user@host>request nic restart resolver name name`

You can restart all NIC resolvers by omitting a resolver name for the **request nic restart resolver** command.

You can also restart a NIC resolver at the slot level.

- Related Topics**
- Stopping a NIC Host on a C Series Controller (SRC CLI) on page 145
  - Restarting the NIC (SRC CLI) on page 145
  - Restarting a NIC Agent (SRC CLI) on page 146
  - Changing NIC Configurations (SRC CLI) on page 147

## Changing NIC Configurations (SRC CLI)

If you change the type of NIC resolution that you use in your network (for example, from the OnePop configuration scenario to the OnePopAllRealms configuration scenario), delete any existing data and specify the scenario name for the new NIC configuration scenario; otherwise, the new NIC configuration may not perform resolutions correctly.

To change the type of NIC resolution that you use in your network:

1. Set the editing level for the CLI to expert.  

```
user@host> set cli level expert
```
2. Disable the NIC:  

```
user@host> disable component nic
```
3. Delete the NIC configuration data for the existing configuration scenario from the directory.  

```
user@host> request nic clear scenario-data
```
4. Navigate to the `[edit slot 0 nic]` hierarchy level.
5. Change the value of **scenario-name** for the local configuration to identify the new configuration scenario. For example:  

```
[edit slot 0 nic]
user@host# set scenario-name OnePopSharedIp
```
6. Return to operational mode, and restart the NIC host.  

```
user@host>request nic slot number restart
```
7. Set the editing level for the CLI to basic.  

```
user@host> set cli level basic
```
8. Configure the new NIC scenario.

- Related Topics**
- Configuring the NIC (SRC CLI) on page 130
  - Configuring Advanced NIC Features on page 143
  - NIC Configuration Scenarios on page 118
  - Configuration Statements for the NIC on page 127
  - Changing NIC Configurations (C-Web Interface)



## CHAPTER 11

# Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrfIp

- Overview of the Network Publisher on page 149
- NIC Document That Maps Subscriber IP Addresses to a JUNOS Interface on page 150
- Configuration Statements for the Network Publisher on page 150
- Before You Configure and Run the Network Publisher on page 151
- Configuring the Network Publisher (SRC CLI) on page 152
- Running the Network Publisher (SRC CLI) on page 157
- Files Used to Test Network Publisher on page 158
- Configuring Information to Test the Network Publisher (SRC CLI) on page 158
- Troubleshooting Network Publisher Operations (SRC CLI) on page 159
- Reviewing the Information Collected from a JUNOS Routing Platform (SRC CLI) on page 160

## Overview of the Network Publisher

---

The network publisher is a NIC component that connects to JUNOS routing platforms and collects information, such as information about system interfaces and VPNs, from IPv4 and IPv6 routing tables. After collecting the information, the network publisher stores this information in the Juniper Networks database for access by the NIC.

Use the network publisher to collect information from JUNOS routing tables for the following configuration scenarios:

- OnePopStaticRouteIp—Resolves an IP address for a subscriber whose traffic enters the network through a JUNOS interface to a reference for the SAE that manages the interface. The Threat Mitigation Application Portal demonstration application relies on this scenario.
- OnePopVrfIp—Resolves an IP address for a subscriber whose traffic enters the network through a VPN configured on a JUNOS interface. This scenario provides support for multiple VPNs that have overlapping IP pools.

You run the network publisher whenever you want to get routing table information from one or more routers; the network publisher does not automatically update configuration information in the directory.

- Related Topics**
- [NIC Document That Maps Subscriber IP Addresses to a JUNOS Interface](#) on page 150
  - [Files Used to Test Network Publisher](#) on page 158
  - [Before You Configure and Run the Network Publisher](#) on page 151
  - [Configuration Statements for the Network Publisher](#) on page 150
  - [Configuring the Network Publisher \(C-Web Interface\)](#)

---

## NIC Document That Maps Subscriber IP Addresses to a JUNOS Interface

NIC stores information about IP pools or networks that map to JUNOS interfaces using routing table information. These files comply with the syntax in the file `/opt/UMC/nic/etc/networkConfig.xsd`. A sample file `/opt/UMC/nic/networkConfig.xml` shows the type of information generated by the network publisher.

- Related Topics**
- [Overview of the Network Publisher](#) on page 149
  - [Reviewing the Information Collected from a JUNOS Routing Platform \(SRC CLI\)](#) on page 160

---

## Configuration Statements for the Network Publisher

Use the following configuration statements to configure the network publisher.

```
slot number network-publisher logger logger-name file {
 filter filter;
 filename filename;
 rollover-filename rollover-filename;
 maximum-file-size maximum-file-size;
}
slot number network-publisher logger logger-name syslog {
 filter filter;
 hostname hostname;
 facility facility;
 format format;
}
slot number network-publisher routers {
 router-release-number router-release-number;
 router-script-version router-script-version;
}
slot number network-publisher routers authentication {
 login-name login-name;
 credentials credentials;
 protocol protocol;
}
slot number network-publisher routers router router-name {
```

```
router-address router-address;
router-release-number router-release-number;
router-script-version router-script-version;
}
slot number network-publisher routers router router-name authentication {
 login-name login-name;
 credentials credentials;
 protocol protocol;
}
slot number network-publisher select {
 route-table-filter route-table-filter;
 route-entry-filter route-entry-filter;
}
slot number network-publisher directory-connection {
 url url;
 principal principal;
 credentials credentials;
 base-dn base-dn;
}
slot number network-publisher routers test-mode {
 enable-file-input;
 input-location input-location;
 enable-file-output;
 output-location output-location;
}
slot number network-publisher routers router router-name test-mode {
 enable-file-input;
 input-location input-location;
 enable-file-output;
 output-location output-location;
}
```

- Related Topics**
- For detailed information about each configuration statement, see the *SRC PE CLI Command Reference*.
  - Overview of the Network Publisher on page 149
  - Before You Configure and Run the Network Publisher on page 151
  - Configuring the Network Publisher (SRC CLI) on page 152

---

## Before You Configure and Run the Network Publisher

Before you configure and run the network publisher:

- Verify the version of the JUNOS software that is running on each JUNOS routing platform.

Typically, all the JUNOS routing platforms should run the same version of the JUNOS software.

- Verify that the C Series Controller can connect to the SAE-managed JUNOS routing platforms.

- Make sure that an SSH (recommended) or a Telnet service is enabled on each router from which the network publisher is to collect interface information.

When you run the network publisher, it connects to a number of JUNOS routing platforms through the configured protocol.

- Identify the routing tables and elements in the routing tables from which you want the network publisher to collect information. Which tables and elements you select depends on the application to use the NIC OnePopStaticRouteIp or the OnePopVrflp configuration scenario.
- Before you run the network publisher, make sure that the NIC is enabled.

**Related Topics**

- Configuring the Network Publisher (SRC CLI) on page 152
- Configuring the Network Publisher (C-Web Interface)
- Starting the NIC (SRC CLI) on page 144
- Overview of the Network Publisher on page 149

---

## Configuring the Network Publisher (SRC CLI)

To configure the network publisher, complete the following tasks:

1. Configuring Local Configuration for the Network Publisher on page 152
2. Configuring Connections Between JUNOS Routing Platforms and the Network Publisher on page 153
3. Configuring Router Authentication for the Network Publisher on page 154
4. Configuring Routing Table Filters for the Network Publisher on page 155
5. Configuring the Connection Between the Network Publisher and the Juniper Networks Database on page 156

### Configuring Local Configuration for the Network Publisher

You configure the network publisher for a slot. There is no shared configuration for the network publisher.

Use the following configuration statements to configure the basic local configuration for the network publisher:

```
slot number network-publisher logger logger-name file {
 filter filter;
 filename filename;
 rollover-filename rollover-filename;
 maximum-file-size maximum-file-size;
}
slot number network-publisher logger logger-name syslog {
 filter filter;
 hostname hostname;
 facility facility;
 format format;
}
```



To set up the basic configuration for the network publisher:

1. From configuration mode, access the configuration statement that specifies the configuration for the network publisher for a slot.

```
[edit]
user@host# edit slot 0 network-publisher
```

2. Configure logging for the network publisher as you do for other SRC components.

## Configuring Connections Between JUNOS Routing Platforms and the Network Publisher

The network publisher connects to the JUNOScript server on a JUNOS routing platform. You can configure connection information for a group of JUNOS routers that use the same version of JUNOScript, and configure information for JUNOS routing platforms that use a different version.

Use the following configuration statements to configure connection information to allow the network publisher to connect to JUNOS routing platforms:

```
slot number network-publisher routers {
 router-release-number router-release-number;
 router-script-version router-script-version;
}
slot number network-publisher routers router router-name {
 router-address router-address;
 router-release-number router-release-number;
 router-script-version router-script-version;
}
```

To configure JUNOScript connection information for the network publisher to connect to JUNOS routing platforms:

1. From configuration mode, access the configuration statement that specifies the configuration for the network publisher for a slot.

```
[edit]
user@host# edit slot 0 network-publisher routers
```

2. Specify the release number of the JUNOS software running on the devices.

```
[edit slot 0 network-publisher routers]
user@host# set router-release-number 8.5R1
```

3. (Optional) Specify the version of JUNOScript running on the JUNOS routing platforms.

```
[edit slot 0 network-publisher routers]
user@host# set router-script-version 1.0
```

4. (Optional) Configure connection information for JUNOS routing platforms that use a different version of the JUNOS to JUNOScript software.

- a. Specify the router name of the router that uses a different version of the software.

```
[edit slot 0 network-publisher routers]
```

```
user@host# set router my-router
```

- b. Configure the IP address of the router.

```
[edit slot 0 network-publisher routers router my-router]
```

```
user@host# set router address 10.10.4..4
```

- c. Specify the release number of the JUNOS software running on the devices.

```
[edit slot 0 network-publisher routers router my-router]
```

```
user@host# set router-release-number 8.5R2
```

- d. Specify the version of JUNOScript running on the JUNOS routing platforms.

```
[edit slot 0 network-publisher routers router my-router]
```

```
user@host# set router-script-version 1.0
```

## Configuring Router Authentication for the Network Publisher

You can configure connection authentication information for a group of JUNOS routing platforms that use the same authentication information, and configure information for JUNOS routing platforms that use a different username and password.



**NOTE:** For the network publisher to access JUNOS routing platforms, configure authentication for all devices or for each specific device.

Use the following configuration statements to configure connection authentication information to allow the network publisher to connect to JUNOS routing platforms:

```
slot number network-publisher routers authentication {
 login-name login-name;
 credentials credentials;
 protocol protocol;
}
slot number network-publisher routers router router-name authentication {
 login-name login-name;
 credentials credentials;
 protocol protocol;
}
```

To configure authentication information for the network publisher to connect to JUNOS routing platforms:

1. From configuration mode, access the configuration statement that specifies the configuration for router authentication.

```
[edit]
```

```
user@host# edit slot 0 network-publisher routers authentication
```

2. Specify the release number of the JUNOS software running on the devices.

```
[edit slot 0 network-publisher routers]
```

```
user@host# set router-release-number 8.5R1
```

3. Specify the protocol to connect to the JUNOS routing platform. We recommend that you use SSH.

```
[edit slot 0 network-publisher routers authentication]
user@host# set protocol ssh
```

4. Specify the username to log into the JUNOS software.

```
[edit slot 0 network-publisher routers authentication]
user@host# set login-name Chris-Bee
```

5. Specify the password for the username.

```
[edit slot 0 network-publisher routers authentication]
user@host# set credentials credentials
```

6. (Optional) Configure authentication information for JUNOS routing platforms that use different authentication information.

- a. Specify the router name.

```
[edit slot 0 network-publisher routers]
user@host# edit router my-router authentication
```

- b. Specify the username to log into the JUNOS software.

```
[edit slot 0 network-publisher routers router my-router authentication]
user@host# set login-name Bee-C
```

- c. Specify the password for the username.

```
[edit slot 0 network-publisher routers router my-router authentication]
user@host# set credentials credentials
```

## Configuring Routing Table Filters for the Network Publisher

The network publisher can collect information from JUNOS IPv4 and IPv6 routing tables. Specify which routing tables the network publisher should include to meet the requirements of your application that uses the NIC OnePopStaticRouteIp or OnePopVrflp configuration scenario.

By default, the network publisher collects information from all IPv4 routing tables, including tables for VPNs, and entries for all protocols. Based on your network configuration, consider which protocols to exclude from the configuration for network publisher.

Use the following configuration statements to identify the routing tables and routing table elements from which to collect information for the network publisher:

```
slot number network-publisher select {
 route-table-filter route-table-filter ;
 route-entry-filter route-entry-filter ;
}
```

To specify the routing tables from which the network publisher collects information:

1. From configuration mode, access the configuration statement that specifies the configuration for the IPv4 and IPv6 routing tables from which the network publisher is to collect information.

```
[edit]
user@host# edit slot 0 network-publisher select
```

2. Specify the routing table from which the network publisher collects information:

```
[edit slot 0 network-publisher select]
user@host# set route-table-filter route-table-filter
```

For example, to select only IPv6 tables:

```
[edit slot 0 network-publisher select]
user@host# set route-table-filter "table-name=*inet6.0"
```

You can use regular expressions to identify routing tables.

3. Specify the element(s) in a routing table:

```
[edit slot 0 network-publisher select]
user@host# set route-entry-filter route-entry-filter
```

For example, to select only those entries that pertain to OSPF advertisements:

```
[edit slot 0 network-publisher select]
user@host# set route-entry-filter "protocol=OSPF"
```

## Configuring the Connection Between the Network Publisher and the Juniper Networks Database

Configure the connection properties that the network publisher uses to connect to the Juniper Networks database. The network publisher can then store information about routing tables from JUNOS routing platforms in the Juniper Networks database.

Use the following configuration statements to configure the connection information that the network publisher uses to connect to the Juniper Networks database:

```
slot number network-publisher directory-connection {
 url url;
 principal principal;
 credentials credentials;
 base-dn base-dn;
}
```

To configure connection information for the Juniper Networks database:

1. From configuration mode, access the configuration statement that specifies the configuration for router authentication.

```
[edit]
user@host# edit slot 0 network-publisher directory-connection
```

2. Specify the URL of the primary Juniper Networks database.

```
[edit slot 0 network-publisher directory-connection]
user@host# set url url
```

3. Specify the distinguished name (DN) that defines the username with which the network publisher accesses the Juniper Networks database, for example `cn = umcadmin, o = umc`.

```
[edit slot 0 network-publisher directory-connection]
user@host# set principal cn=umcadmin,o=umc
```

4. Specify the password with which the network publisher accesses the Juniper Networks database; for example:

```
[edit slot 0 network-publisher directory-connection]
user@host# set credentials admin123
```

5. (Optional) Specify the DN of the subtree in the database that stores the router data; for example `o = Network, o = umc`:

```
[edit slot 0 network-publisher directory-connection]
user@host# set base-dn o=Network,o=umc
```

- Related Topics**
- Before You Configure and Run the Network Publisher on page 151
  - Configuring System Logging (SRC CLI)
  - Configuring a Component to Store Log Messages in a File (SRC CLI)
  - Running the Network Publisher (SRC CLI) on page 157
  - Overview of the Network Publisher on page 149

---

## Running the Network Publisher (SRC CLI)

You run the network publisher each time you want to collect information from routing tables on JUNOS routing platforms.

Before you run the network publisher, make sure that:

- The network publisher is configured.
- The NIC is enabled.

To run the network publisher:

- From operational mode, run one of the following commands:

```
user@host> request network-publisher execute
```

```
user@host> request network-publisher slot 0 execute
```

- Related Topics**
- Before You Configure and Run the Network Publisher on page 151
  - Configuring the Network Publisher (SRC CLI) on page 152
  - Starting the NIC (SRC CLI) on page 144

- Overview of the Network Publisher on page 149
- Files Used to Test Network Publisher on page 158

## Files Used to Test Network Publisher

---

You can configure the network publisher to use files to test a configuration or to troubleshoot network publisher operation.

Network publisher supports the following types of files:

- Input files—Use to test a configuration before routes to the NIC are available or before VPNs are configured. You can also use input files to set up a test configuration for demonstration purposes.
- Output files—Use to view the information collected from the router to see whether the network publisher is collecting the information you expect.

You must enable the network publisher to use files. Although you can specify a directory location for these files at the advanced editing level, we recommend that you use the default filenames:

- Input file—`/opt/UMC/nic/var/sample/junos/rt/router—name_1.xml`
- Output file for a specific router—`/opt/UMC/nic/var/junos/rt/router—name_1.xml`

### Related Topics

- Overview of the Network Publisher on page 149
- Configuring Information to Test the Network Publisher (SRC CLI) on page 158
- Reviewing the Information Collected from a JUNOS Routing Platform (SRC CLI) on page 160

## Configuring Information to Test the Network Publisher (SRC CLI)

---

You can use an input file to verify that the network publisher is collecting information as configured or to set up a demonstration for an application.

To configure the network publisher to use an input file:

1. Enable the network publisher to use an input file for all routers or for a specific router.

Sample syntax for all routers:

```
[edit slot 0 network-publisher routers test-mode]
user@host# set enable-file-input
```

Sample syntax to collect information for a router named my-router:

```
[edit slot 0 network-publisher routers router my-router test-mode]
user@host# set enable-file-input
```

2. Run the network publisher.

```
user@host> request network-publisher execute
```

- Related Topics**
- Configuring Information to Test the Network Publisher (C-Web Interface)
  - Overview of the Network Publisher on page 149
  - Files Used to Test Network Publisher on page 158
  - Troubleshooting Network Publisher Operations (SRC CLI) on page 159

## Troubleshooting Network Publisher Operations (SRC CLI)

**Problem** The network publisher is not collecting the expected data.

- Solution**
1. Make sure that the network publisher can connect to the configured routers.
  2. Make sure that authentication is configured correctly for the network publisher and on the router.
  3. Verify the configuration for the network publisher.

```
[edit slot 0 network-publisher]
user@host# show
directory-connection {
 url ldap://127.0.0.1:389;
 base-dn o=Network,o=UMC;
 principal cn=umcadmin,o=umc;
 credentials *****;
}
select {
}
logger log1 {
 file {
 filter /debug-;
 filename var/log/netpub_debug.log;
 rollover-filename var/log/netpub_debug.alt;
 maximum-file-size 2000000000;
 }
}
logger log2 {
 file {
 filter /info-;
 filename var/log/netpub_info.log;
 rollover-filename var/log/netpub_info.alt;
 maximum-file-size 2000000000;
 }
}
logger log3 {
 file {
 filter /error-;
 filename var/log/netpub_error.log;
 rollover-filename var/log/netpub_error.alt;
 maximum-file-size 2000000000;
 }
}
routers {
 router-release-number 7.6R1;
 authentication {
```

```
login-name admin2;
credentials *****;
}
router elf {
 address 10.227.7.115;
}
router giant {
 address 10.227.7.124;
}
}
```

4. Configure the network publisher to use an input file to ensure that the network publisher is collecting information as configured. Modify the content of the input file to reflect the router information.

See “Configuring Information to Test the Network Publisher (SRC CLI)” on page 158

5. Configure the network publisher to use an output file, and review the file.

See “Reviewing the Information Collected from a JUNOS Routing Platform (SRC CLI)” on page 160

- Related Topics**
- Before You Configure and Run the Network Publisher on page 151
  - Configuring the Network Publisher (SRC CLI) on page 152
  - Overview of the Network Publisher on page 149

## Reviewing the Information Collected from a JUNOS Routing Platform (SRC CLI)

**Purpose** Review information that the network publisher collects from a JUNOS routing platform.

- Action**
1. Enable an output file to collect information from all routers or for a specific router.

Sample syntax for all routers:

```
[edit slot 0 network-publisher routers test-mode]
user@host# set enable-file-output
```

Sample syntax to collect information for a router named my-router:

```
[edit slot 0 network-publisher routers router my-router test-mode]
user@host# set enable-file-output
```

2. Run the network publisher.  

```
user@host> request network-publisher execute
```
3. Use FTP to transfer the file from the C Series Controller to another system; then open the file on the remote system and examine the file content.

- Related Topics**
- Overview of the Network Publisher on page 149
  - Files Used to Test Network Publisher on page 158
  - Troubleshooting Network Publisher Operations (SRC CLI) on page 159



- Specifying Filenames and URLs
- Reviewing the Information Collected from a JUNOS Routing Platform (C-Web Interface)



## CHAPTER 12

# Configuring Applications to Communicate with an SAE

- Overview of NIC Proxy Configuration on page 163
- Before You Configure a NIC Proxy on page 164

### Overview of NIC Proxy Configuration

---

You configure applications to communicate with network information collector (NIC) hosts. A NIC host can be local within an application, or external to the application. For Java applications, you also configure NIC proxies as part of an application.

For a number of SRC components, such as the SRC Volume-Tracking Application (SRC VTA) and the Dynamic Service Activator, you can configure the NIC proxy for the application from the SRC CLI. For other applications, such as the sample residential portal, you configure the NIC proxy in a property file. If you configure a NIC proxy from a property file, the fields are the same as the fields that appear at the CLI. When you develop and test SRC components that use a NIC, you can configure a NIC proxy stub to take the place of the NIC host.

For more information about NIC proxies, see “Locating Subscriber Management Information” on page 111.

#### Related Topics

- Configuring Resolution Information for a NIC Proxy (SRC CLI) on page 166
- Changing the Configuration for the NIC Proxy Cache (SRC CLI) on page 168
- Before You Configure a NIC Proxy on page 164
- Configuration Statements for NIC Proxies on page 165
- Configuring a NIC Proxy for NIC Replication (SRC CLI) on page 169
- Removing the NIC Proxies on page 180

## Before You Configure a NIC Proxy

---

Before you configure a NIC proxy, you should have a good understanding of:

- NIC resolution
- NIC data types
- How NIC proxies work

See “Locating Subscriber Management Information” on page 111, “Overview of the NIC Resolution Process” on page 181, and “Overview of NIC Proxy Configuration” on page 163.



**NOTE:** You cannot configure a local NIC host when the NIC is running on a C Series Controller.

The values that you configure for a NIC proxy depend on the particular application; for example, you must specify the type of data used for the key and the type of data used for the value for each application.

Before you configure a NIC proxy for an application, obtain the following information from the system manager who maintains the NIC configuration for NIC hosts:

- The name of the resolver that the application uses.
- The type of key the application will provide to the NIC host.
- The type of value the NIC host is to return.
- Whether or not the application will use a local NIC host.
- If the application does not use a local NIC host:
  - The size of the NIC proxy cache.
  - The groups to be listed for NIC host selection. These groups provide NIC replication.

### Related Topics

- Configuring a NIC Proxy (C-Web Interface)
- Configuring Resolution Information for a NIC Proxy (SRC CLI) on page 166
- Changing the Configuration for the NIC Proxy Cache (SRC CLI) on page 168
- Instantiating a Configuration Manager on page 176
- Configuration Statements for NIC Proxies on page 165

# Configuring SRC Applications to Communicate with an SAE (SRC CLI)

- Configuration Statements for NIC Proxies on page 165
- Configuring Resolution Information for a NIC Proxy (SRC CLI) on page 166
- Changing the Configuration for the NIC Proxy Cache (SRC CLI) on page 168
- Configuring a NIC Proxy for NIC Replication (SRC CLI) on page 169
- Configuring NIC Test Data (SRC CLI) on page 171

## Configuration Statements for NIC Proxies

---

Use the following configuration statements to configure a NIC proxy for SRC components. You access these statements from the hierarchy for a component, such as:

- [edit shared acp configuration]
- [edit shared sae configuration]

```
nic-proxy-configuration name {
}

nic-proxy-configuration name resolution {
 resolver-name resolver-name;
 key-type key-type;
 value-type value-type;
 expect-multiple-values;
 constraints constraints;
}

nic-proxy-configuration name cache {
 cache-size cache-size;
 cache-cleanup-interval cache-cleanup-interval;
 cache-entry-age cache-entry-age;
}

nic-proxy-configuration name nic-host-selection {
 groups groups;
 selection-criteria (roundRobin | randomPick | priorityList);
}

nic-proxy-configuration name nic-host-selection blacklisting {
 try-next-system-on-error;
```

```
number-of-retries-before-blacklisting number-of-retries-before-blacklisting;
blacklist-retry-interval blacklist-retry-interval;
}
```

Use the following statements to configure a NIC proxy stub for SRC components. You access these statements from the hierarchy for a component, such as:

- [edit shared dsa configuration]
- [edit shared sae configuration]

```
nic-proxy-configuration name test-nic-bindings {
 use-test-bindings;
}

nic-proxy-configuration name test-nic-bindings key-values name {
 value;
}
```

- Related Topics**
- Before You Configure a NIC Proxy on page 164
  - For detailed information about each configuration statement, see *SRC PE CLI Command Reference*.
  - Configuring Resolution Information for a NIC Proxy (SRC CLI) on page 166
  - Changing the Configuration for the NIC Proxy Cache (SRC CLI) on page 168
  - Configuring a NIC Proxy for NIC Replication (SRC CLI) on page 169

---

## Configuring Resolution Information for a NIC Proxy (SRC CLI)

Use the following statements to configure resolution information for a NIC proxy:

```
nic-proxy-configuration name resolution {
 resolver-name resolver-name;
 key-type key-type;
 value-type value-type;
 expect-multiple-values;
 constraints constraints;
}
```

To configure resolution information for a NIC proxy:

1. From configuration mode, access the configuration statement that specifies the NIC proxy configuration.

```
[edit]
user@host# component-hierarchy nic-proxy-configuration name resolution
```

For example:

```
[edit]
user@host# edit shared sae configuration nic-proxy-configuration ip resolution
```

2. Specify the NIC resolver that this NIC proxy uses.

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set resolver-name resolver-name
```

This resolver must be the same as one that is configured on the NIC host. For example:

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set resolver-name /realms/ip/A1
```

3. Specify the NIC data type that the key provides for the NIC resolution.

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set key-type key-type
```

For example:

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set key-type ip
```

To qualify data types, enter a qualifier within parentheses after the data type; for example, to specify username as a qualifier for the key LoginName:

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set key-type LoginName (username)
```

4. Specify the type of value to be returned in the resolution for the application that uses the NIC proxy.

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set value-type value-type
```

For example:

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set value-type SaeId
```

5. (Optional) If the key can have more than one value, specify that the key can have multiple corresponding values.

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set expect-multiple-values
```

6. (Optional. Available at the Advanced editing level.) If the application provides a constraint in the resolution request, specify the data type for the constraint. The constraint represents a condition that must or may be satisfied before the next stage of the resolution process can proceed.

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set constraints constraints
```

#### Related Topics

- Before You Configure a NIC Proxy on page 164
- Changing the Configuration for the NIC Proxy Cache (SRC CLI) on page 168
- Configuring a NIC Proxy for NIC Replication (SRC CLI) on page 169

- Configuration Statements for NIC Proxies on page 165
- Overview of NIC Proxy Configuration on page 163

## Changing the Configuration for the NIC Proxy Cache (SRC CLI)

---

You can modify cache properties for the NIC proxy to optimize the resolution performance for your network configuration and system resources. Typically, you can use the default settings for the cache properties. The configuration statements are available at the Advanced editing level.

Use the following configuration statements to change values for the NIC proxy cache:

```
nic-proxy-configuration name cache {
 cache-size cache-size;
 cache-cleanup-interval cache-cleanup-interval;
 cache-entry-age cache-entry-age;
}
```

To configure the cache for a NIC proxy:

1. From configuration mode, access the configuration statement that specifies the NIC proxy configuration.

```
[edit]
user@host# component-hierarchy nic-proxy-configuration name cache
```

For example:

```
[edit]
user@host# edit shared sae configuration nic-proxy-configuration ip cache
```

2. Specify the maximum number of keys for which the NIC proxy retains data.

```
[edit shared sae configuration nic-proxy-configuration ip cache]
user@host# set cache-size cache-size
```

If you decrease the cache size or disable the cache while the NIC proxy is running, the NIC proxy removes entries in order of descending age until the cache size meets the new limit.

3. Specify the time interval at which the NIC proxy removes expired entries from its cache.

```
[edit shared sae configuration nic-proxy-configuration ip cache]
user@host# set cache-cleanup-interval cache-cleanup-interval
```

4. Specify how long an entry remains in the cache.

```
[edit shared sae configuration nic-proxy-configuration ip cache]
user@host# set cache-entry-age cache-entry-age
```

**Related Topics**   • Before You Configure a NIC Proxy on page 164



- Configuring Resolution Information for a NIC Proxy (SRC CLI) on page 166
- Configuring a NIC Proxy for NIC Replication (SRC CLI) on page 169
- Configuration Statements for NIC Proxies on page 165
- Overview of NIC Proxy Configuration on page 163

## Configuring a NIC Proxy for NIC Replication (SRC CLI)

Typically, you configure NIC replication to keep the NIC highly available. You configure NIC host selection to specify the groups of NIC hosts to be contacted to resolve a request, and to define how the NIC proxy handles NIC hosts that the proxy is unable to contact. The configuration statements are available at the Advanced editing level.

Use the following configuration statements to configure NIC host selection for a NIC proxy:

```

nic-proxy-configuration name nic-host-selection {
 groups groups;
 selection-criteria (roundRobin | randomPick | priorityList);
}

nic-proxy-configuration name nic-host-selection blacklisting {
 try-next-system-on-error;
 number-of-retries-before-blacklisting number-of-retries-before-blacklisting;
 blacklist-retry-interval blacklist-retry-interval;
}

```

To configure a NIC proxy to use NIC replication:

1. From configuration mode, access the configuration statement that specifies the NIC proxy configuration.

```

[edit]
user@host# component-hierarchy nic-proxy-configuration name nic-host-selection

```

For example:

```

[edit]
user@host# edit shared sae configuration nic-proxy-configuration ip
nic-host-selection

```

2. Specify the list of groups of NIC hosts that the NIC proxy can contact for resolution requests. Use commas to separate the group names.

```

[edit shared sae configuration nic-proxy-configuration ip nic-host-selection]
user@host# set groups groups

```

For example

```

[edit shared sae configuration nic-proxy-configuration ip nic-host-selection]
user@host# set groups [group1 group2]

```

3. If you configure more than one group, specify the selection criteria that the NIC proxy uses to determine which NIC host to contact.

```
[edit shared sae configuration nic-proxy-configuration ip nic-host-selection]
user@host# set selection-criteria (roundRobin | randomPick | priorityList)
```

where:

- roundRobin—NIC proxy selects NIC hosts in a fixed, cyclic order. The NIC proxy always selects the next host in the list.
- randomPick—NIC proxy selects NIC hosts randomly from the list.
- priorityList—NIC proxy selects NIC hosts according to their assigned priorities in the list. If the host with the highest priority in the list is not available, the NIC proxy tries the host with the next-highest priority, and so on.

Priorities are defined by the order in which you specify the groups. You can change the order of NIC hosts in the list by using the **insert** command.

4. Access the configuration statement that specifies the NIC proxy configuration for blacklisting—the process of handling nonresponsive NIC hosts.

```
[edit shared sae configuration nic-proxy-configuration ip nic-host-selection]
user@host# edit blacklisting
[edit shared sae configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
```

5. Specify whether or not the NIC proxy should contact the next specified NIC host if a NIC host is determined to be unavailable.

```
[edit shared sae configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
user@host# set try-next-system-on-error
```

6. (Optional) Change the number of times the NIC proxy tries to communicate with a NIC host before the NIC proxy stops communicating with the NIC host for a period of time. The default is 3.

```
[edit shared sae configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
user@host# set number-of-retries-before-blacklisting
number-of-retries-before-blacklisting
```

7. (Optional) Change the interval at which the NIC proxy attempts to connect to an unavailable NIC host. The default is 15 seconds.

```
[edit shared sae configuration nic-proxy-configuration name nic-host-selection
blacklisting]
user@host# set blacklist-retry-interval blacklist-retry-interval
```

#### Related Topics

- Before You Configure a NIC Proxy on page 164
- Configuring Resolution Information for a NIC Proxy (SRC CLI) on page 166

- Changing the Configuration for the NIC Proxy Cache (SRC CLI) on page 168
- Configuration Statements for NIC Proxies on page 165
- Overview of NIC Proxy Configuration on page 163

## Configuring NIC Test Data (SRC CLI)

To test a resolution without NIC, you can configure a NIC proxy stub to take the place of the NIC. The NIC proxy stub comprises a set of explicit mappings of data keys and values in the NIC proxy configuration. When the SAE (or another SRC component configured to use a NIC proxy stub) passes a specified key to the NIC proxy stub, the NIC proxy stub returns the corresponding value. When you use a NIC proxy stub, no NIC infrastructure is required.

For example, you can specify a subscriber's IP address that is associated with a particular SAE. When the SRC component passes this IP address to the NIC proxy stub, the NIC proxy stub returns the corresponding SAE.

To use the NIC proxy stub for the SAE:

1. In configuration mode, navigate to the NIC proxy configuration and specify the type of key you want to map to a value.

```
[edit shared sae configuration nic-proxy-configuration name]
user@host# set resolution key-type key-type
```

For example, to specify the key ip for the ip NIC proxy configuration:

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set resolution key-type ip
```

2. Enable a NIC proxy stub for a resolution.

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings user-test-bindings
```

3. Specify the values of the keys for testing. These statements are available at the Expert CLI editing level.

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings key-values name value
```

where:

- ***name***—Indicates the NIC data value for the proxy.
- ***value***—Specifies a value for the NIC data type.

For example, to set up a login name to IP mapping for login name jane@virneo.com to the IP address 192.0.2.30:

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings key-values jane@virneo.com 192.0.2.30
```

For example, to set up an IP to SAE ID mapping for IP address 190.0.2.30 to SAE ID identified by the URL for the CORBA IOR `corbaloc::10.227.7.145:8801/SAE`:

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings key-values 192.0.2.30
corbaloc::10.20.7.145:8801/SAE
```



---

**NOTE:** The SAE writes the value of the CORBA IOR to the `var/run` directory. The IP address in the `corbaloc` URL can be adjusted to the IP address or DNS name of the SAE.

---

You can use the key **ANY\_KEY** to match any key for any key type. For example, if you want all IP addresses to resolve to the same SAE:

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings key-values ANY_KEY
corbaloc::10.20.7.145:8801/SAE
```

- Related Topics**
- Planning a NIC Implementation on page 117
  - NIC Configuration Scenarios on page 118
  - High Availability for NIC on page 114
  - Configuring NIC Test Data (C-Web Interface)

# Developing Applications That Use NIC

- External Application Requirements for NIC on page 173
- External Non-Java Applications That Use NIC on page 173
- Creating a NIC Locator to Include with a Non-Java Application on page 174
- External Java Applications That Use NIC on page 175
- Developing a Java Application to Communicate with a NIC Proxy on page 176
- Updating Information About Address Pools on page 180

## External Application Requirements for NIC

---

If you write an external application to use NIC to perform a resolution, you can include NIC functionality in one of the following ways:

- For non-Java applications, use the interface module `NicAccess`, an IDL file that provides access to the NIC locator feature. The NIC locator can resolve the value of one or more keys.
- For Java applications, include the NIC proxy client libraries to use NIC in client/server mode.
- For Java applications, include the NIC proxy client libraries and the NIC host client libraries to use NIC in local host mode.

### Related Topics

- External Non-Java Applications That Use NIC on page 173
- External Java Applications That Use NIC on page 175
- Creating a NIC Locator to Include with a Non-Java Application on page 174

## External Non-Java Applications That Use NIC

---

If you write an application in a language other than Java, you can use the NIC access interface module, a simplified CORBA interface, to perform one or more resolutions. By using this interface you can access through CORBA NIC locators, NIC proxies that run within the NIC host. The configuration properties for NIC locators are similar to those for NIC proxies in applications such as aggregate services and the sample residential portal.

- Related Topics**
- For information about the NIC access interface module, see the API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.
  - External Application Requirements for NIC on page 173
  - External Java Applications That Use NIC on page 175
  - Creating a NIC Locator to Include with a Non-Java Application on page 174

---

## Creating a NIC Locator to Include with a Non-Java Application

A NIC locator provides the same functionality as a NIC proxy, but is designed to work with non-Java applications.

You use the NIC access interface module to include NIC locators with your application by compiling the IDL file with your application files.

To use the NIC access interface module to create NIC locators:

1. Connect to the directory.
2. Obtain a CORBA reference to the NIC access interface from one of the following:
  - The access IOR provided in the directory in the dynamic configuration DN under the hostname—typically, *host/demohost*.
  - A corbaloc URL in the format:  
`corbaloc::<host>:8810/Access`

3. From the NIC access interface module, obtain a NIC locator, as identified by `NicFeature`. For example:

```
feature = access.getLocatorFeature(nicNameSpace); //nicNameSpace example "
/nicLocators/ip"
```

In the NIC configuration scenarios, the syntax for a NIC locator is `/nicLocators/<NIC key type>` where.

- **nicLocators**— Specifies all of the NIC locators in a NIC host.
  - **<NIC key type>**— Specifies the type of data that the key provides for the NIC resolution, such as ip, login, DN.
4. Search for the key. For example:  

```
feature.lookupSingle(NicLocatorKey key) //NicLocatorKey is coming from the IDL
```

- Related Topics**
- For information about the NIC access interface module, see the API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.
  - External Java Applications That Use NIC on page 175

- External Application Requirements for NIC on page 173
- External Non-Java Applications That Use NIC on page 173

## External Java Applications That Use NIC

---

If you write an external Java application that interacts with a NIC, include NIC libraries in the application. These libraries are for NIC proxies and local NIC hosts. These libraries are located in the SDK+AppSupport+Demos+Samples.tar.gz on the Juniper Networks Web site at: <https://www.juniper.net/support/csc/swdist-erx/src.html>. You can locate the files in their *SDK/lib/nic* directory.

Typically, each NIC resolution process requires one NIC proxy. For example, the OnePopLogin sample data includes two resolution processes:

- Mapping of a subscriber's IP address to the subscriber's login name
- Mapping of the subscriber's login name to the SAE reference

An application that uses both these resolution processes would require two NIC proxies.

The NIC proxy provides a simple Java interface, the NIC application programming interface (API). You configure the NIC proxy to communicate with one resolver. For efficiency if you use NIC in client/server mode, the NIC proxy caches the results of resolution requests so it can respond to future requests for the same key without contacting the resolver.

The SRC software includes a factory interface, the NIC factory, to allow applications to instantiate, access, and remove NIC proxies. It also includes JAR files for NIC client and NIC host libraries.

You must configure an application to communicate with a NIC proxy.

If you are using Java Runtime Environment (JRE) 1.3 or higher, you must include in your application the Java archive (JAR) files, available in the SDK+AppSupport+Demos+Samples.tar.gz file on the Juniper Networks Web site at: <https://www.juniper.net/support/csc/swdist-erx/src.html>. The files are located in the */SDK/lib/* directory.

- Related Topics**
- For more information about the API calls, see the online documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.
  - External Application Requirements for NIC on page 173
  - External Non-Java Applications That Use NIC on page 173
  - Creating a NIC Locator to Include with a Non-Java Application on page 174

## Developing a Java Application to Communicate with a NIC Proxy

---

Configuration tasks that use the API calls to communicate with the NIC proxy are:

- Instantiating a Configuration Manager on page 176
- Passing a Reference to the Configuration Manager to the NIC Factory on page 176
- Instantiating the NIC Factory Class on page 176
- Initializing Logging on page 177
- Instantiating the NIC Proxy on page 177
- Managing a Resolution Request on page 178
- Deleting Invalid Results from the NIC Proxy's Cache on page 179
- Removing the NIC Proxies on page 180

### Instantiating a Configuration Manager

The application must instantiate a configuration manager.

To enable the application to instantiate a configuration manager to obtain a NIC instance from the NIC factory:

- Call one of the following methods:
  - For some applications (other than Web applications), in which you must define the system property `-DConfig.bootstrapFilename`, you can call the following method:  

```
ConfigMgr configMgr = ConfigMgrFactory.getConfigMgr();
```
  - For Web applications, you can instantiate the configuration manager as follows:  

```
ConfigMgr configMgr = ConfigMgrFactory.getConfigMgr(properties);
```
  - `properties`—`java.util.Properties` object, typically the bootstrap file, which contains all the configuration properties for the NIC proxy.

### Passing a Reference to the Configuration Manager to the NIC Factory

To pass a reference to the configuration manager to the NIC factory class:

- Call the following method in the application:  

```
NicFactory.setConfigManager(configMgr);
```

### Instantiating the NIC Factory Class

The way you instantiate the NIC factory depends on the object request broker (ORB) configuration:

- If the NIC proxy uses the default ORB, call the following method in the application:  

```
NicFactory nicFactory = NicFactory.getInstance();
```



This code instantiates a new NIC factory. Unless the `NicFactory.destroy` method has been called, subsequent calls to this method will return the instantiated NIC factory.

- If the NIC proxy does not use the default ORB, call the following method:

```
NicFactory.initialize(props);
NicFactory nicFactory = NicFactory.getInstance();
```

- `props`—`java.util.Properties` object, which contains the ORB properties for the NIC proxy. For example, if the NIC proxy uses JacORB but JacORB is not the default ORB, the ORB properties are:

```
org.omg.CORBA.ORBClass=org.jacorb.orb.ORB
org.omg.CORBA.ORBSingletonClass=org.jacorb.orb.ORBSingleton
```

This code will instantiate a new NIC factory using the specified ORB. Unless the application has called the `NicFactory.destroy` method, subsequent calls to the `getInstance()` method will return the instantiated NIC factory. However, if the application has called the `destroy()` method, it must recall the `initialize()` method before it can call the `getInstance()` method.

For information about the `NicFactory.destroy` method, see “Removing the NIC Proxies” on page 180.

## Initializing Logging

You must initialize logging only if you want to view the logging information produced by the NIC proxy.

To enable the application to initialize logging:

- Call the following method:

```
Log.init(configMgr, configNameSpace);
```

- `configMgr`—Instance of the configuration manager, the value returned from the `getConfigMgr()` method
- `configNameSpace`—String that specifies the configuration namespace where you defined the logging properties
  - If you define the logging properties in the bootstrap file, specify the root namespace, `"/"`.

```
Log.init(configMgr, "/");
```

- If you define the logging properties in the directory, specify the namespace relative to the property `Config.net.juniper.smgmt.lib.config.staticConfigDN`, which you configure in the bootstrap file.

```
Log.init(configMgr, "/Applications/Quota");
```

## Instantiating the NIC Proxy

To enable the application to instantiate a NIC proxy:

- Call the following method:

```
NIC nicProxy = nicFactory.getNicComponent(nicNameSpace, configMgr)
```

Alternatively, if the expected data value (specified for the property `nic.value` in the NIC proxy configuration) is an SAE reference, you can call the following method:

```
SaeLocator nicProxy = nicFactory.getSaeLocator(nicNameSpace, configMgr);
```

- `nicFactory`—Instance of the NIC factory
- `nicNameSpace`—String that specifies the configuration namespace where you defined the properties for the NIC proxy
  - If you define the NIC properties in the bootstrap file, specify the root namespace, `"/"`.

```
NIC nicProxy = nicFactory.getNicComponent("/", configMgr)
```

- If you define the properties in the directory, specify the namespace relative to the property `Config.net.juniper.smgmt.lib.config.staticConfigDN`, which you specified in the bootstrap file.

```
NIC nicProxy = nicFactory.getNicComponent("/Applications/Quota", configMgr)
```

- `configMgr`—Instance of the configuration manager, the value returned from the `getConfigMgr()` method

## Managing a Resolution Request

To enable the application to submit a resolution request and obtain the associated values:

1. Construct a `NicKey` object to enable the application to pass the data key to the NIC proxy:

```
NicKey nicKey = new NicKey(stringKey);
```

- `stringKey`—Data key for which you want to find corresponding values.

For the syntax of allowed data types, see “Overview of the NIC Resolution Process” on page 181.

2. If the resolution process specifies constraints that you wish to provide in the resolution request, add them to the `NicKey` object:

```
NicKey.addConstraint(constName, constValue);
```

- `constName`—Name of the constraint.

For the allowed data types and their syntax, see “Overview of the NIC Resolution Process” on page 181.

- `constValue`—Specific value of the constraint.

For the allowed syntax for the data types, see “Overview of the NIC Resolution Process” on page 181.

3. Call a method that starts the resolution process.

For example, you can call a method specified in the NIC interface:

```
NicValue val = nicProxy.lookupSingle(nicKey);
```

Alternatively, if the expected data value is an SAE reference, you can call the following method:

```
Saeld saeld = nicProxy.lookupSae(nicKey);
```

4. Call the `getValue` method to access the string representation of the data value obtained by the NIC proxy.

```
String val=val.getValue();
```

Alternatively, if the expected data value is an SAE reference:

```
String val=saeld.getValue();
```

5. (Optional) Call a method to get intermediate values obtained during a resolution.
  - Call the `getIntermediateValue` method if the application expects only one value. This method takes the name of a data type and returns as a string the first value it finds.

```
String getIntermediateValue(String dataTypeName){};
}
```

For information about data types, see “Overview of the NIC Resolution Process” on page 181.

- Call the `getIntermediateValues` or `getAllIntermediateValues` method if the application expects multiple values. These methods take the name of a data type and return values as follows:
  - The `getIntermediateValues` method returns a list of values as a string array.

```
String[] getIntermediateValues(String dataTypeName){};
```

- For information about data types, see “Overview of the NIC Resolution Process” on page 181
- The `getAllIntermediateValues` method returns a map of all intermediate values for the request. The key for the map is the name of the network data type, and the value of the map is a string array of the intermediate values.

```
Map getAllIntermediateValues();
```

## Deleting Invalid Results from the NIC Proxy's Cache

If the application receives an exception when using values that the NIC proxy returned for a specific key, it must inform the NIC proxy to delete this entry from its cache.

To enable the application to inform the NIC proxy to delete an entry from its cache:

- Call the following method:

```
nicProxy.invalidateLookup(nicKey, nicValue);
```

- `nicKey`—Data key that you want to remove from the cache

- `nicValue`—Optional data value that corresponds to this key

If the application passes a null data value to the NIC proxy, the NIC proxy removes all the values associated with the data key from its cache.

## Removing the NIC Proxies

Make sure that before your application shuts down, it removes the NIC proxy instances to release resources for other software processes.

To remove one NIC proxy instance:

- Call the following method:

```
NicProxy.destroy();
```

To remove all NIC proxy instances, call the following method:

```
NicFactory.destroy();
```

## Updating Information About Address Pools

---

If you associate an existing address pool with an interface and you do not want to wait for this new information to be propagated based on the Cache Entry Age property of the NIC proxy or the Event Life Expectancy property of the agents, then you must manually clear the NIC proxy cache.

To clear the NIC proxy cache when an application is deployed in a J2EE container that supports Java Management Extension (JMX) software, do one of the following:

- Use the `NicProxyMgmt` MBean.
- Restart the application.
- Restart the application server.

### Related Topics

- [Deleting Invalid Results from the NIC Proxy's Cache on page 179](#)
- [Removing the NIC Proxies on page 180](#)
- [Passing a Reference to the Configuration Manager to the NIC Factory on page 176](#)
- [External Non-Java Applications That Use NIC on page 173](#)

## CHAPTER 15

# NIC Resolution Process

- Overview of the NIC Resolution Process on page 181
- NIC Data Types on page 182
- Constraints as NIC Data Types on page 184

### Overview of the NIC Resolution Process

---

Because NIC can process all types of network data, you must use different resolution processes for different types of data mappings to maximize the performance of the NIC configuration. Resolving data requests consumes significant resources.

Table 12 on page 181 shows the resolutions that the components in the NIC configuration scenarios perform. For customized types of resolutions, contact Juniper Networks Professional Services.

**Table 12: Available NIC Resolutions**

| Key                                              | Value                   |
|--------------------------------------------------|-------------------------|
| Subscriber's IP address (JUNOS routing platform) | SAE reference           |
| Subscriber's IP address                          | Subscriber's login name |
| Subscriber's IP address                          | SAE reference           |
| Subscriber's login name                          | SAE reference           |
| Subscriber's username                            | SAE reference           |
| Access DN                                        | SAE reference           |

### NIC Realms

Each resolution process and the resolvers that perform that process are defined by a *realm*—a group of resolvers that perform a series of resolution tasks to provide a mapping from a specified key to a specified data type. For example, the sample data provided for the NIC includes a realm called *dn* in which the resolution process takes an *access*

subscriber's distinguished name (DN) as the key and returns a reference to the SAE managing this subscriber as the value.

A set of hosts in a NIC can support multiple realms. Similarly, the agents in a NIC can support more than one realm. However, you can assign a resolver only to one realm.

A NIC host can support NIC resolvers for multiple realms. Consequently, you can simplify the NIC configuration and minimize the use of network resources by limiting the number of NIC hosts in your NIC configuration. NIC hosts can also handle multiple NIC resolvers in the same realm. In this case, when a NIC host receives a request, it chooses a NIC resolver as follows:

1. It identifies the NIC resolvers that are available to process the request.
2. If multiple NIC resolvers are available, it obtains a cost value associated with the resolution process from each resolver and selects the resolver that has the lowest cost value.

## Key to Value Resolution

A resolution process typically defines several transitions or *roles*, with each transition resolving a NIC key to a value. For example, the resolution process to identify the SAE that manages a particular subscriber based on that subscriber's IP address involves the following roles:

1. Given the IP address, determine the IP address pool.
2. From the IP address pool, determine the VR.
3. From the VR, determine the SAE that manages that VR.

A role specifies the types of data with which it works. NIC supports a number of data types, including one that lets you add an identifier to other data types to let you specify different values for one data type.

For information about NIC data types, see "NIC Data Types" on page 182 and "Constraints as NIC Data Types" on page 184.

**Related Topics**

- Managing a Resolution Request on page 178

---

## NIC Data Types

The NIC supports the data types that appear in the following list. You can qualify these data types by adding an identifier to:

- Distinguish between different instances of a data type in a resolution scenario.
- Provide information about a data type to clarify the use of that data type in a resolution.

### *AnyString*

- Generic data type to represent the information that you want to collect.
- Value—Alphanumeric characters
- Guidelines—You can qualify this data type with an identifier to provide information about the type of data that AnyString represents.
- Example—My(IP), My(Vr)

**Dn**

- DN of an access.
- Value—DN
- Example—*accessName=PrimaryAccess, enterpriseName=juniper, ou=Sunnyvale, retailerName=VPNprovider, o=Users, o=umc*

**Domain**

- Domain name.
- Value—Name of a domain
- Example—Example.net

**Enterprise**

- DN of an enterprise.
- Value—DN
- Example—*enterpriseName=juniper, ou=Sunnyvale, retailerName=VPNprovider, o=Users, o=umc*

**Router**

- Name of router.
- Value—Text string
- Example—router1

**Interface**

- Name of a router's interface. Can include a virtual routing forwarding identifier Vrfid). If a Vrfid is present, the DSA passes it to the SAE in an assignedIp request. The SAE uses the Vrfid to support IP addresses that may be the same across different VRFs.
- Value—`<interfaceName>/<ID>@<vrName>@ <routerName>`  
`<interfaceName>#<vrId>@vrName@routerName`
- Example—FastEthernet4/1.0/4@boston@router1  
fastEthernet4/1.0#vpn\_a@boston@router1

***InterfaceId***

- Identifier of an interface.
- Value—<intfIndex>@<routerName>
- Example—4@router1

***Ip***

- Subscriber's IP address.
- Value—IP address
- Example—192.0.2.10

***IpPool***

- IP address pool.
- Value—Range of IP addresses enclosed in square brackets and parentheses
- Guidelines—If you enter an IP address that includes a value greater than 255 in one octet of the address, that part of the address is masked to fit the eight bits.
- Example—([192.0.2.0 192.0.2.255])

***SaeId***

- SAE reference.
- Value—CORBA interoperable object reference (IOR) for SAE
- Example—IOR:0000000000000002438444C3A736...

***Vr***

- Name of the virtual router.
- Value—<vrName>@<routerName>
- Example—vr1@router1

---

**Constraints as NIC Data Types**

---

Constraints are data types that a resolver uses when it executes a role. You can define:

- Multiple constraints for a role—Software performs an OR operation to determine whether the constraint is met.
- Multiple data types in a constraint—Software performs an AND operation to determine whether the multiple constraints are met.

Constraints can be either mandatory or optional. If a constraint is mandatory and the resolver for the role does not receive an appropriate value in the data request, the resolver must obtain the constraint value from other NIC resolvers. However, if a constraint is



optional and the resolver for the role does not receive an appropriate value in the data request, the resolver can execute its role without the constraint value. In this case, the resolver may obtain multiple values for the data key, and the NIC host responds to the NIC proxy as follows:

- If the request is for multiple results, the host provides all the results.
- If the request is for one result and the resolution process returns different results, the host returns an error message.
- If the resolution process returns multiple instances of the same result, the resolver provides only one result.

For example, if you want to obtain an SAE reference for a subscriber's IP address, you could define the following roles:

1. From the IP address, determine the VR (mandatory constraint IpPool).
2. From the VR, determine the SAE that manages that VR.

Because the first step has a mandatory constraint, the resolver for this role must use the IP pool supplied in the request, or obtain the IP pool from another resolver that determines IP pools from IP addresses. So you must define an extra step at the start of the resolution process:

1. From the IP address, determine the IP pool.
2. From the IP address, determine the VR (mandatory constraint IpPool).
3. From the VR, determine the SAE that manages that VR.

- Related Topics**
- Overview of the NIC Resolution Process on page 181
  - NIC Data Types on page 182
  - Managing a Resolution Request on page 178



## CHAPTER 16

# NIC Configuration Scenarios

- Overview of NIC Configuration Scenarios on page 187
- OnePop Scenario on page 188
- OnePopPcmm Scenario on page 190
- OnePopDynamicIp Scenario on page 192
- OnePopSharedIp Scenario on page 194
- OnePopStaticRouteIp Scenario on page 196
- OnePopVrfIp Scenario on page 199
- OnePopAcctId Scenario on page 201
- OnePopLogin Scenario on page 203
- OnePopLoginPull Scenario on page 205
- OnePopPrimaryUser on page 206
- OnePopDnSharedIp Scenario on page 208
- OnePopAllRealms Scenario on page 212
- MultiPop Scenario on page 216

## Overview of NIC Configuration Scenarios

---

The NIC configuration scenarios in the sample data provide resolutions for a variety of network configurations.

Each NIC scenario includes two types of configuration:

- Centralized—A single host configuration for use with NIC replication. In a centralized configuration all agents and resolvers reside on one host. The name of this host is DemoHost.
- Distributed—A multiple host configuration in which agents and resolvers are distributed among more than one host. This type of configuration is designed for use with NIC host redundancy. In most cases, the hosts are named OnePopH1 (a host in a pop) and OnePopBO (a host in a back office).

The best way to view the sample data is with the NIC Web Admin tool.

For a summary of the NIC configuration scenarios included in the sample data, see “NIC Configuration Scenarios” on page 118 .

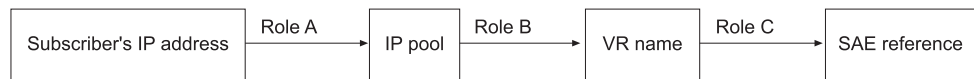
- Related Topics**
- Overview of NIC Proxy Configuration on page 163
  - Before You Configure the NIC on page 129
  - Configuration Statements for the NIC on page 127
  - Configuring the NIC (SRC CLI) on page 130

## OnePop Scenario

The OnePop scenario illustrates a configuration that supports one POP. The realm for this configuration accommodates the situation in which IP address pools are configured locally on each VR. The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value.

Figure 15 on page 188 shows the resolution graph for this realm.

**Figure 15: Resolution Process for ip Realm**



g014923

The following agents collect information for resolvers in this realm:

- Directory agent PoolVr collects and publishes information about the mappings of IP address pools to VRs.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The OnePop sample provides two host configurations: a centralized configuration and a distributed configuration. The OnePop Centralized configuration also provides an example of NIC host redundancy.

## Centralized Configuration

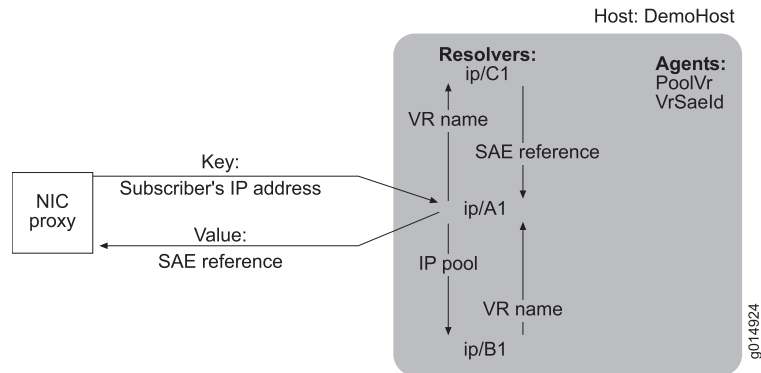
In this configuration, single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of actions occurs:

1. The host passes the IP address to resolver A1.
2. Resolver A1 obtains an IP pool for the IP address and forwards the request to resolver B1.
3. Resolver B1 obtains a VR name for the IP pool and returns the VR name to resolver A1.
4. Resolver A1 forwards the VR name to resolver C1.

5. Resolver C1 obtains an SAE reference for the VR and returns the VR identity to resolver A1.
6. Resolver A1 passes the SAE reference to its host.
7. The host returns the SAE reference to the NIC proxy.

Figure 16 on page 189 shows the interactions of the NIC components for this realm.

**Figure 16: OnePop Centralized Configuration**

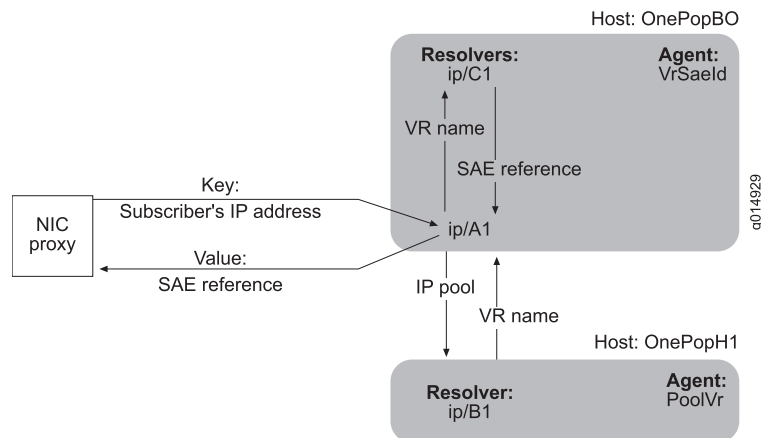


## Distributed Configuration

In this configuration, the agents and resolvers are distributed among several hosts. When the NIC proxy sends a subscriber's IP address to host OnePopBO, the components execute the same actions as they do in the centralized configuration.

Figure 17 on page 189 illustrates the interactions of the NIC components for this realm.

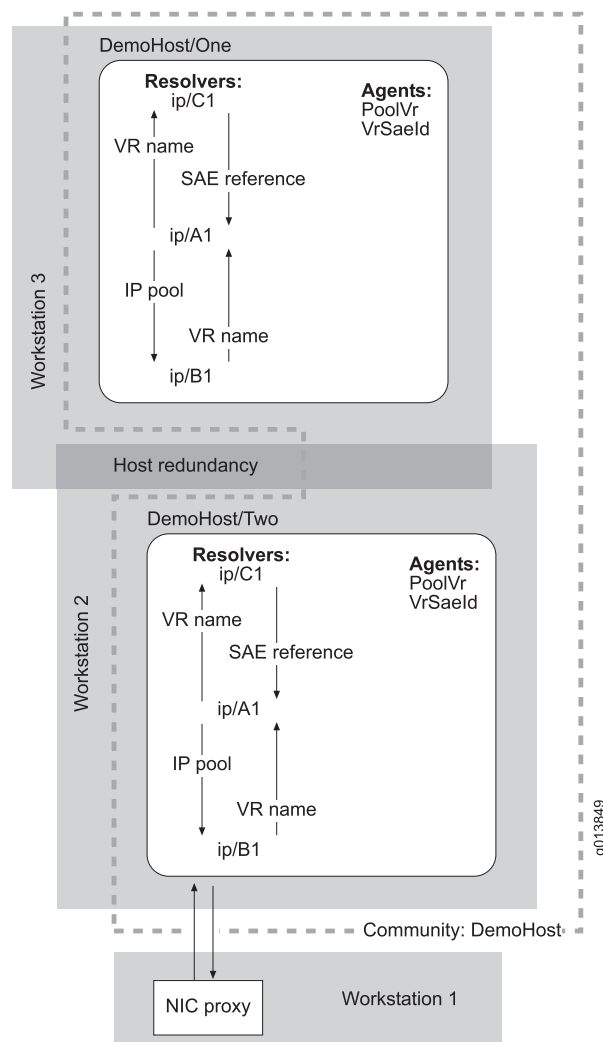
**Figure 17: OnePop Distributed Configuration**



## Redundancy

This sample data includes host redundancy for the centralized configuration. The hosts DemoHost/One and DemoHost/Two, which are installed on different machines, provide host redundancy. These hosts form the community DemoHost, which does not include a monitor.

Figure 18: Redundancy for OnePop Centralized Configuration



- Related Topics**
- Overview of NIC Configuration Scenarios on page 187
  - Configuring a NIC Scenario (SRC CLI) on page 134

## OnePopPcmm Scenario

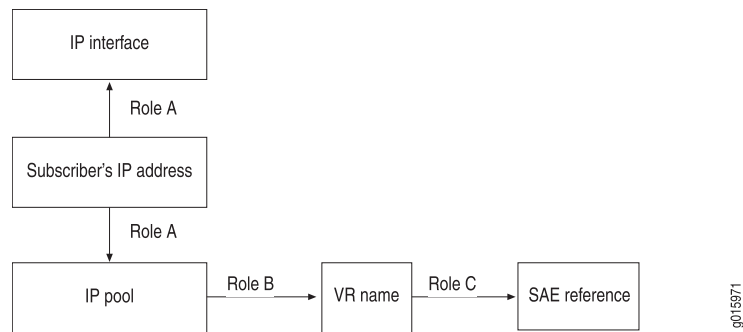
This scenario is similar to the OnePop configuration scenario. It illustrates a configuration in which an assigned subscriber IP address managed by a network device such as a cable modem termination system (CMTS) device resolves to a reference to the SAE managing this subscriber. In this situation, the SAE acts as an application manager and interacts with the CMTS through a policy server.

The OnePopPcmm configuration scenario supports a PacketCable Multimedia Specification (PCMM) environment in which you use the assigned IP subscriber method to log in subscribers and in which you use the NIC to determine the subscriber's SAE. The

realm for this configuration accommodates the situation in which IP pools are configured locally on each application manager group object. These IP pools represent an IP pools-managed policy decision point (PDP) group for one or more CMTS devices.

Figure 19 on page 191 shows the resolution graph for this realm.

**Figure 19: Resolution Process for Pcomm\_am Realm**



This scenario uses the same agents as the OnePop scenario. For the OnePopPcomm configuration scenario, the agent collects information from the application manager object instead of the virtual router entry. A virtual router name is generated in the format "default"@<pdpGroup>.

The OnePopPcomm scenario provides two host configurations: a centralized configuration and a distributed configuration.

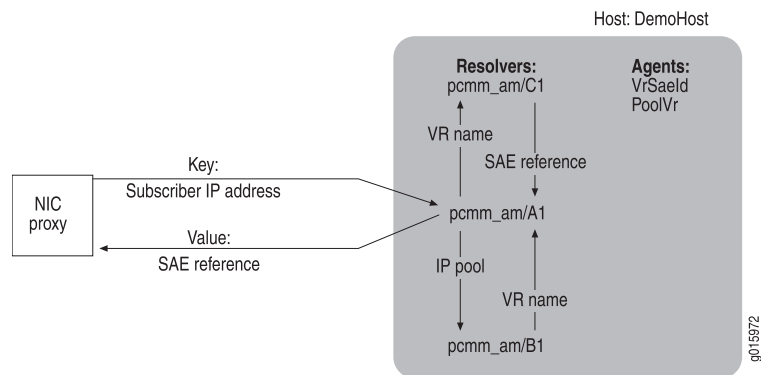
### Centralized Configuration

In this configuration, the single host DemoHost supports all agents and resolvers. When a NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of actions occurs:

1. The host passes an assigned subscriber IP address resolver A1.
2. Resolver A1 obtains the IP pool name and the interface name, and forwards the request to resolver B1.
3. Resolver B1 obtains the VR name for the IP pool name and interface name, and returns the VR name to resolver A1.
4. Resolver A1 forwards the VR name to resolver C1.
5. Resolver C1 obtains an SAE reference for the VR and returns it to resolver A1.
6. Resolver A1 passes the SAE reference to its host.
7. The host returns the SAE reference to the NIC proxy.

Figure 20 on page 192 show the interactions of the NIC components for this realm.

Figure 20: OnePopPcmm Centralized Configuration

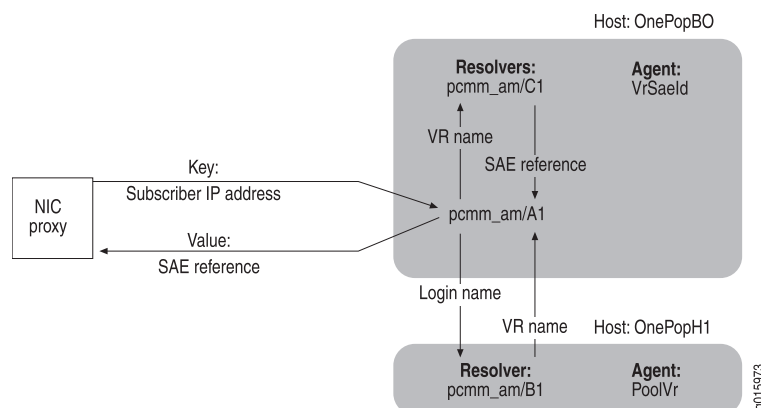


## Distributed Configuration

In this configuration, the agents and resolvers are distributed among two hosts. When the NIC proxy sends a subscriber's IP address to host OnePopBO, the components execute the same actions as they do in the centralized configuration.

Figure 21 on page 192 illustrates the interactions of the NIC components for this realm.

Figure 21: OnePopPcmm Distributed Configuration



- Related Topics**
- Overview of NIC Configuration Scenarios on page 187
  - Configuring a NIC Scenario (SRC CLI) on page 134

## OnePopDynamicIp Scenario

This scenario illustrates a configuration that is very similar to the OnePop scenario. The realm for this configuration accommodates the situation in which IP address pools are configured locally on each virtual router object. The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value.

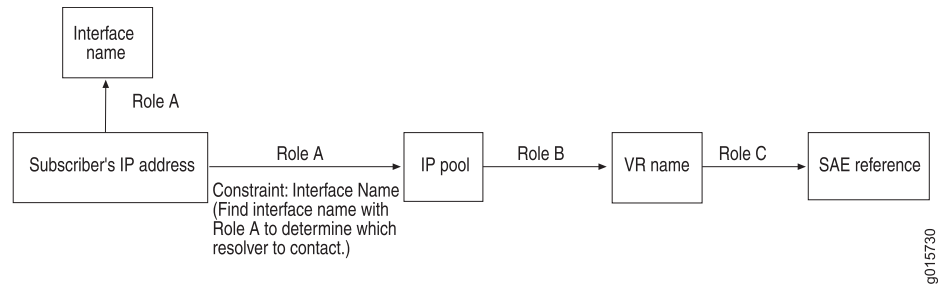
The scenario supports a configuration scenario for a PacketCable Multimedia Specification (PCMM) environment in which you use the assigned IP subscriber method to log in



subscribers, and use the NIC to determine the subscriber's SAE. In this scenario, the SAE acts as a combined application manager and policy server; it directly manages CMTS devices.

Figure 22 on page 193 shows the resolution graph for this realm.

**Figure 22: Resolution Process for dynamicIp Realm**



The following agents collect information for resolvers in this realm:

- Directory agent PoolVr collects and publishes information about the mappings of IP address pools to VRs.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The OnePopDynamicIp scenario provides two host configurations: a centralized configuration and a distributed configuration.

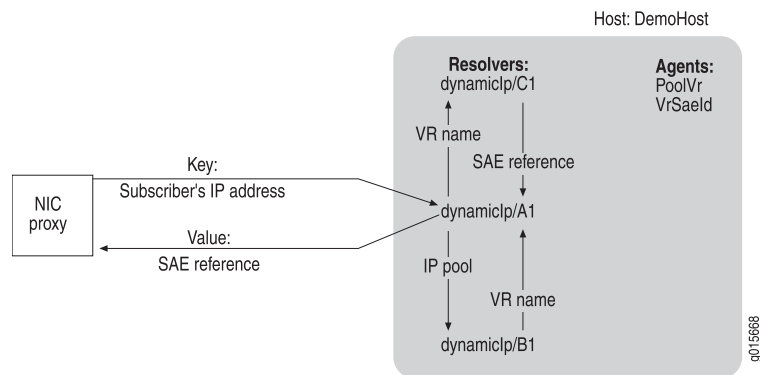
### Centralized Configuration

In this configuration, single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of actions occurs:

1. The host passes the IP address to resolver A1.
2. Resolver A1 obtains an IP pool name and interface name for the IP address, and forwards the request to resolver B1.
3. Resolver B1 obtains a VR name for the IP pool name and interface name, and returns the VR name to resolver A1.
4. Resolver A1 forwards the VR name to resolver C1.
5. Resolver C1 obtains an SAE reference for the VR and returns the VR identity to resolver A1.
6. Resolver A1 passes the SAE reference to its host.
7. The host returns the SAE reference to the NIC proxy.

Figure 23 on page 194 illustrates the interactions of the NIC components for this realm.

Figure 23: OnePopDynamicIp Centralized Configuration

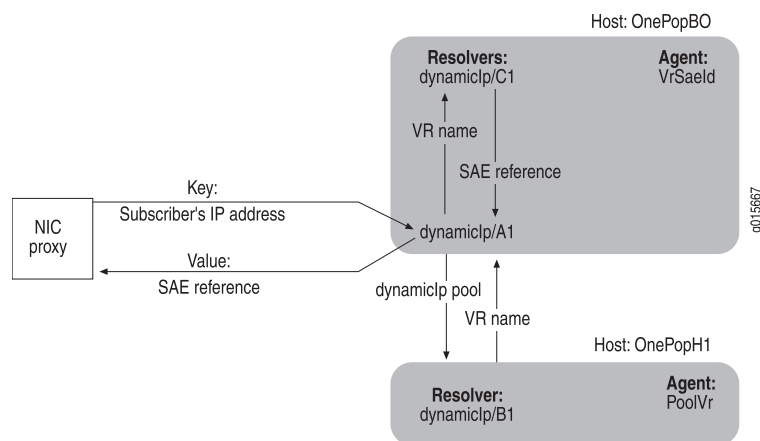


## Distributed Configuration

In this configuration, the agents and resolvers are distributed among several hosts. When the NIC proxy sends a subscriber's IP address to host OnePopBO, the components execute the same actions as they do in the centralized configuration.

Figure 24 on page 194 illustrates the interactions of the NIC components for this realm.

Figure 24: OnePopDynamicIp Distributed Configuration



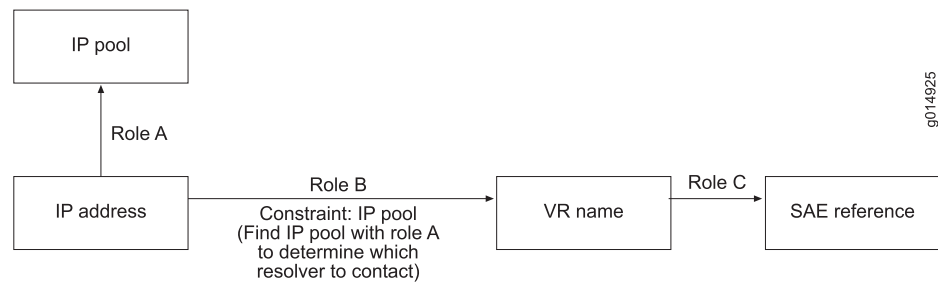
- Related Topics**
- Overview of NIC Configuration Scenarios on page 187
  - Configuring a NIC Scenario (SRC CLI) on page 134

## OnePopSharedIp Scenario

This scenario illustrates a configuration that is very similar to the OnePop scenario. However, the realm for this configuration accommodates the situation in which IP address pools are shared by VRs in the same POP. The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value.

Figure 25 on page 195 shows the resolution graph for this realm.

Figure 25: Resolution Process for sharedIp Realm



g014925

The following agents interact with resolvers in this realm:

- SAE plug-in agent IpVr collects and publishes information about the mappings of IP addresses to VRs.
- Directory agent PoolVr collects and publishes information about the IP address pools used by the VRs in a POP. Because the IP address pools are shared between VRs, this agent discards information about VRs.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The OnePopSharedIP scenario provides two host configurations: a centralized configuration and a distributed configuration.

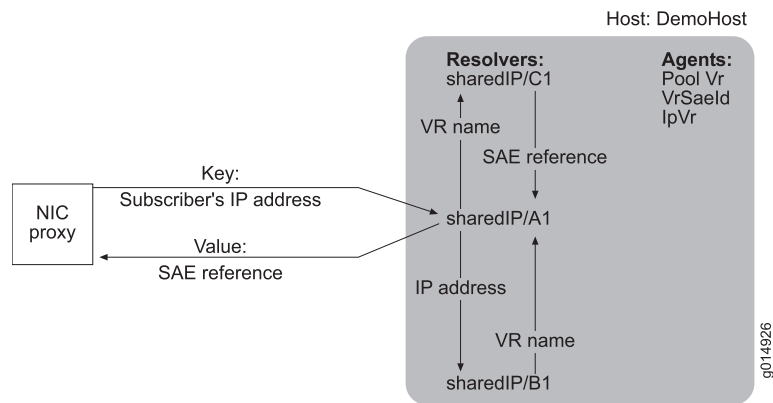
### Centralized Configuration

In this configuration, single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of events occurs:

1. The host passes the IP address to resolver A1.
2. Resolver A1 obtains an IP pool for the IP address.
3. Resolver A1 forwards the IP address and the IP pool to resolver B1.
4. Resolver B1 obtains a VR name for the IP address and returns the VR name to resolver A1.
5. Resolver A1 forwards the VR name to resolver C1.
6. Resolver C1 obtains an SAE reference for the VR and returns the SAE reference to resolver A1.
7. Resolver A1 passes the SAE reference to its host.
8. The host returns the SAE reference to the NIC proxy.

Figure 26 on page 196 shows the interactions of the NIC components for this realm.

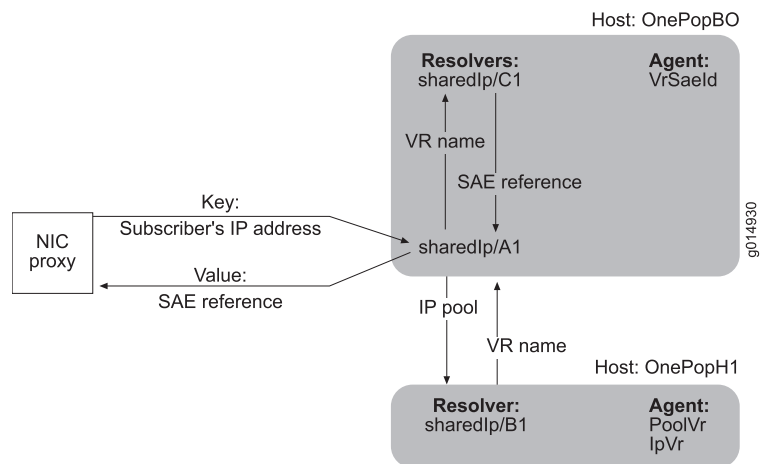
Figure 26: OnePopSharedIP Centralized Configuration



## Distributed Configuration

In this configuration, the agents and resolvers are distributed among several hosts. When the NIC proxy sends a subscriber's IP address to the host OnePopBO, the resolvers execute the same actions as they do in the centralized configuration. Figure 27 on page 196 illustrates the interactions of the NIC components for this realm.

Figure 27: OnePopSharedIP Distributed Configuration



- Related Topics**
- Overview of NIC Configuration Scenarios on page 187
  - Configuring a NIC Scenario (SRC CLI) on page 134

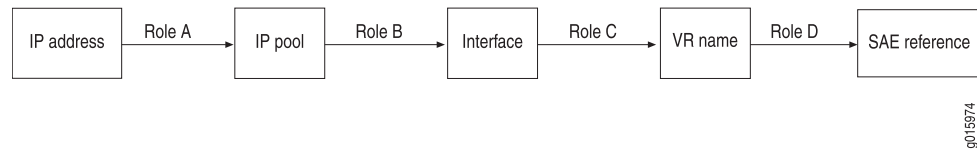
## OnePopStaticRouteIp

The OnePopStaticRouteIp configuration scenario for NIC resolves an assigned IP address for a subscriber whose traffic enters the network through an interface on a JUNOS routing platform to a reference for the SAE that manages the interface. The realm for this configuration accommodates the situation in which the network publisher component gathers interface information for the JUNOS routing platforms. The resolution process

takes a subscriber's IP address as a key and returns a reference to the SAE that manages the interface.

Figure 28 on page 197 shows the resolution graph for this realm.

**Figure 28: Resolution Process for the StaticRoutelp Realm**



The following agents collect information for resolvers in this realm:

- Directory agent PoolInterface collects and publishes information about the mappings of IP address pools to interfaces.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The agents obtain information from the interfaceConfiguration attribute of the EdgeRouter entry in the directory and read an XML document that conforms to the networkConfig.xsd schema. If this scenario is used with a different router type, you can edit the XML document.

For information about the XML document, see “External Application Requirements for NIC” on page 173.

The OnePopStaticRoutelp scenario provides two host configurations: a centralized configuration and a distributed configuration.

## Centralized Configuration

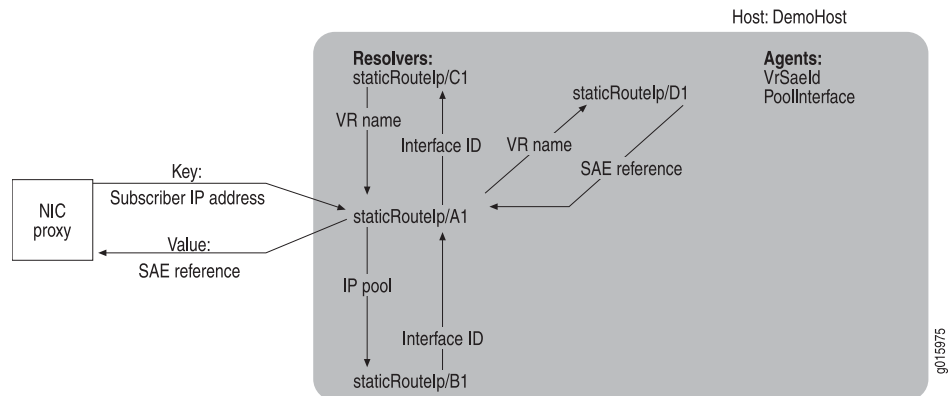
In this configuration, the single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of events occurs:

1. The host passes the subscriber's IP address to resolver A1.
2. Resolver A1 obtains an IP pool for the IP address.
3. Resolver A1 forwards the IP pool name to Resolver B1.
4. Resolver B1 obtains the interface ID for the IP pool and returns this value to resolver A1.
5. Resolver A1 forwards the interface ID to Resolver C1.
6. Resolver C1 resolves the interface ID to the VR name and returns the VR name to resolver A1.
7. Resolver A1 forwards the VR name to resolver D1.
8. Resolver D1 obtains a reference for the SAE managing the VR and returns the SAE reference to resolver A1.

9. Resolver A1 passes the SAE reference to its host.
10. The host returns the SAE reference to the NIC proxy.

Figure 29 on page 198 shows the interactions of the NIC components for this realm.

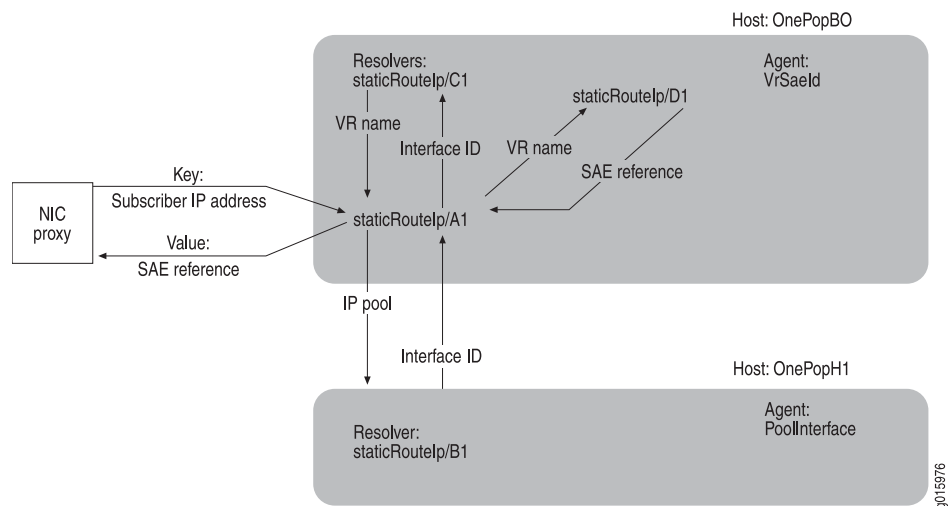
### Figure 29: OnePopStaticRouteIp Centralized Configuration



## Distributed Configuration

In this configuration, the agents and resolvers are distributed among two hosts. When a NIC proxy sends a subscriber IP address to host OnePopBO, the resolvers execute the same actions as they do in the centralized configuration. Figure 30 on page 198 illustrates the interactions of the NIC components for this realm.

### Figure 30: OnePopStaticRouteIp Distributed Configuration



## Related Topics

- Overview of NIC Configuration Scenarios on page 187
- Configuring a NIC Scenario (SRC CLI) on page 134

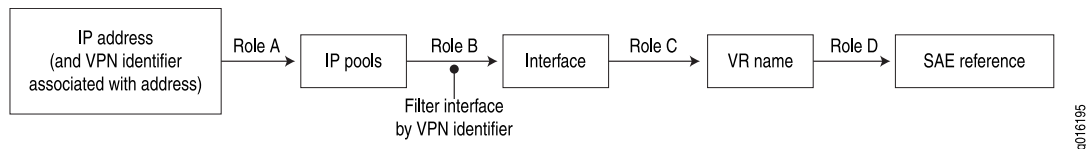
## OnePopVrflp Scenario

The OnePopVrflp configuration scenario for NIC resolves an assigned IP address for a subscriber to IP pools or network whose traffic enters the network through an interface on a JUNOS routing platform to a reference for the SAE that manages the interface. The realm for this configuration utilizes routing information collected by the network publisher from particular JUNOS routing platforms. The resolution process takes a subscriber's IP address as a key and returns a reference to the SAE that manages the interface.

This configuration scenario is very similar to the OnePopStatic Routelp scenario. During resolution, the OnePopVrflp scenario filters interfaces the VPN identifier of the VPN that carries subscriber traffic.

Figure 31 on page 199 shows the resolution graph for this realm.

**Figure 31: Resolution Process for the Vrflp Realm**



The following agents collect information for resolvers in this realm:

- Directory agent PoolInterface collects and publishes information about the mappings of IP address pools to interfaces.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The agents obtain information from the interfaceConfiguration attribute of the EdgeRouter entry in the directory and read an XML document that conforms to the networkConfig.xsd schema. If this scenario is used with a different router type, you can edit the XML document.

For information about the XML document, see “Files Used to Test Network Publisher” on page 158.

The OnePopVrflp scenario provides two host configurations: a centralized configuration and a distributed configuration.

### Centralized Configuration

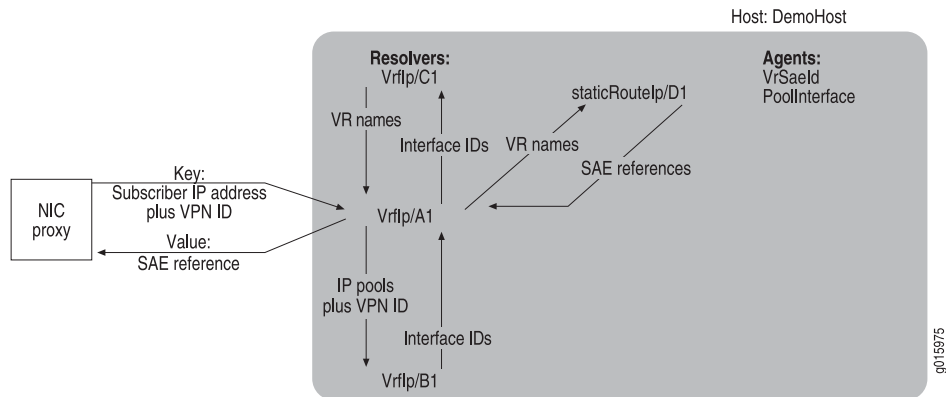
In this configuration, the single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of events occurs:

1. The host passes the subscriber's IP address and VPN ID to resolver A1.
2. Resolver A1 obtains all IP pools that match the IP address.
3. Resolver A1 forwards the IP pool names and VPN ID to Resolver B1.

4. Resolver B1 obtains the all interface IDs for the IP pools and filters all interfaces that match the VPN ID.
5. Resolver A1 forwards the interface IDs to Resolver C1.
6. Resolver C1 resolves the interface IDs to the VR name and returns the VR name to resolver A1.
7. Resolver A1 forwards the VR names to resolver D1.
8. Resolver D1 obtains references for the SAEs managing the VRs and returns the SAE reference to resolver A1.
9. Resolver A1 passes the SAE references to its host.
10. The host returns the SAE references to the NIC proxy.

Figure 32 on page 200 shows the interactions of the NIC components for this realm.

**Figure 32: OnePopVrflp Centralized Configuration**

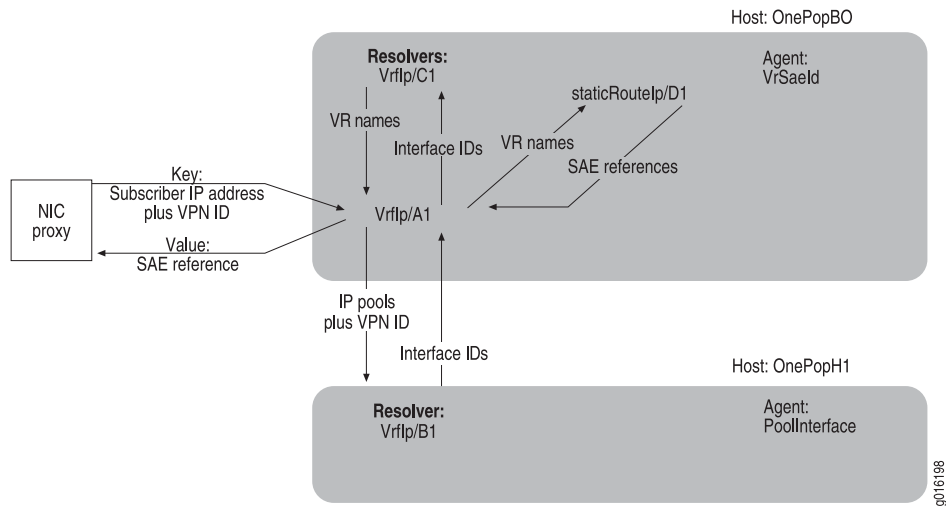


## Distributed Configuration

In this configuration, the agents and resolvers are distributed among two hosts. When a NIC proxy sends a subscriber IP address to host OnePopBO, the resolvers execute the same actions as they do in the centralized configuration. Figure 33 on page 201 illustrates the interactions of the NIC components for this realm.



Figure 33: OnePopStaticRouteIp Distributed Configuration

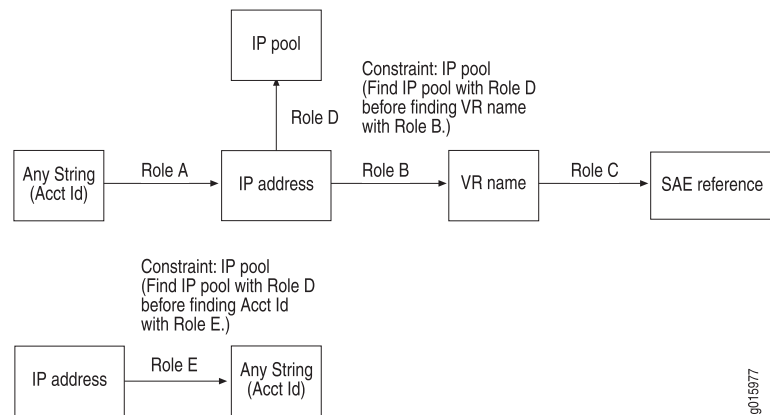


## OnePopAcctId Scenario

This scenario illustrates a configuration in which subscribers have an accounting ID, as defined by the LDAP attribute `accountingUserId` or the plug-in attribute `PA_ACCOUNTING_ID`. The realms for this configuration accommodate two independent resolution processes, which can be used by the SRC Volume-Tracking Application (SRC VTA).

Figure 34 on page 201 shows the resolution graphs for this realm.

Figure 34: Resolution Process for acctId Realm



The following agents collect information for resolvers in this realm:

- Directory agent `PoolVr` collects and publishes information about the mappings of IP address pools to VRs.
- Directory agent `VrSaeld` collects and publishes information about the mappings of virtual routers and the mappings between virtual routers and SAEs.

- SAE plug-in agent AcctIdIp collects and publishes information about the mappings of accounting IDs of subscribers to subscriber IP addresses.
- SAE plug-in agent IpAcctId collects and publishes information about the mappings of subscriber IP addresses to accounting IDs.

The OnePopAcctId scenario provides one host for a centralized configuration. In this configuration the single host DemoHost supports all agents and resolvers. Two NIC proxies are associated with the configuration. One NIC proxy (called acct-sae in this description) submits accounting IDs, and another NIC proxy (called addr-acct in this description) submits subscribers' IP addresses.

When the NIC proxy sends an accounting ID to host DemoHost, the following sequence of events occurs:

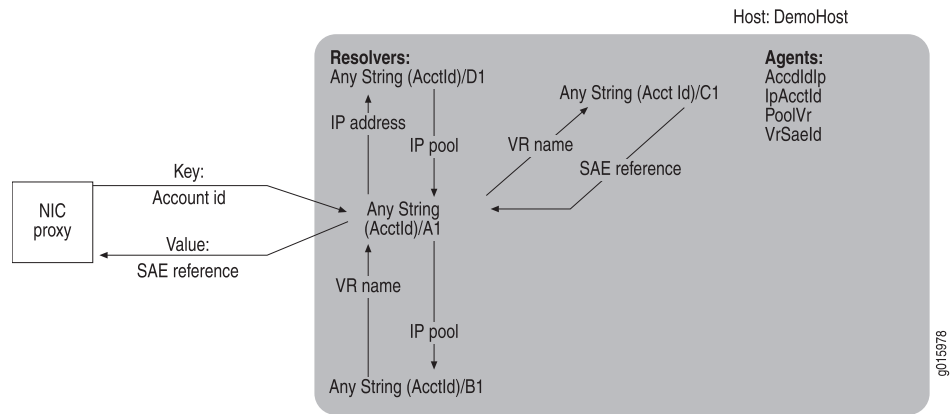
1. The host passes the subscriber's accounting ID to resolver A1.
2. Resolver A1 obtains an IP address for the account ID.
3. Resolver A1 forwards the IP address to Resolver D1.
4. Resolver D1 obtains the IP pool for the IP address and returns it to Resolver A1.
5. Resolver A1 forwards the IP address and IP pool to Resolver B1.
6. Resolver B1 obtains the VR name and return it to resolve A1.
7. Resolver A1 forwards the VR name to resolver C1.
8. Resolver C1 obtains the SAE reference for the VR name and returns it to resolver A1.
9. Resolver A1 passes the SAE reference to its host.
10. The host returns the SAE reference to the NIC proxy acct-sae.

When the NIC proxy sends an IP address to host DemoHost, the following sequence of events occurs:

1. The host passes the subscriber's IP address to resolver A1.
2. Resolver A1 forwards the IP address to resolver D1.
3. Resolver D1 obtains the IP pool for the IP address and returns it to resolver A1.
4. Resolver A1 forwards the IP address and IP pool to resolver C1.
5. Resolver C1 obtains the accounting ID for the IP address and associated IP pool and returns the accounting Id to resolver A1.
6. Resolver A1 passes the accounting ID to its host.
7. The host returns the accounting ID to the NIC proxy addr-acct.

Figure 35 on page 203 illustrates the interactions of the NIC components for this realm.

Figure 35: OnePopAcctId Centralized Configuration



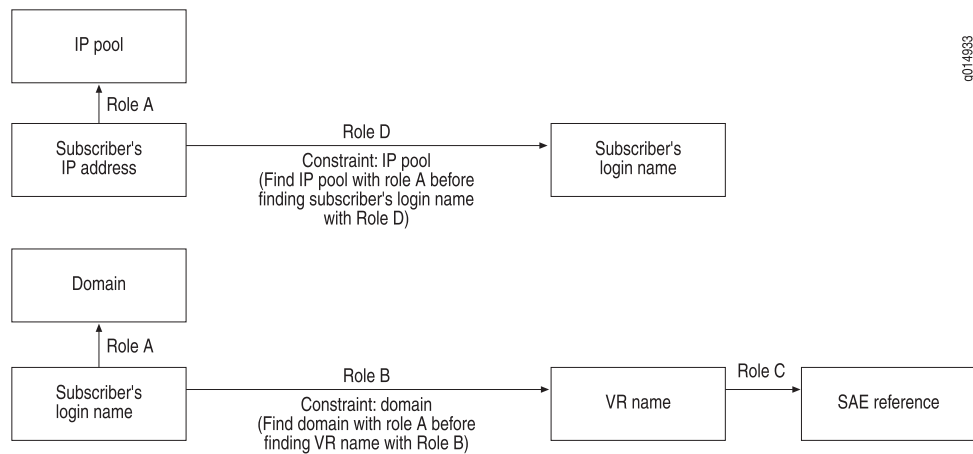
- Related Topics**
- Overview of NIC Configuration Scenarios on page 187
  - Configuring a NIC Scenario (SRC CLI) on page 134

## OnePopLogin Scenario

This scenario illustrates a configuration that is very similar to the OnePop scenario. The realm for this configuration accommodates two independent resolution processes, which are used by the SRC Volume Tracking Applications (SRC VTAs) and may be used for other purposes.

Figure 36 on page 203 shows the resolution graphs for this realm.

Figure 36: Resolution Processes login Realm



The following agents interact with resolvers in this realm:

- SAE plug-in agent **IpLoginName** collects and publishes information about the mappings of IP addresses to login names.
- SAE plug-in agent **LoginNameVr** collects and publishes information about the mappings of login names to VRs.

- Directory agent Pool collects and publishes information about the IP address pools used by the VRs in a POP. The agent uses the information about the IP address pools to determine which resolver to communicate with, rather than communicating with all resolvers that are running role D.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The OnePopLogin scenario provides two host configurations: a centralized configuration and a distributed configuration.

## Centralized Configuration

In this configuration, single host DemoHost supports all agents and resolvers. Two NIC proxies are associated with this NIC configuration; one NIC proxy (called NIC proxy 1 in this documentation) submits subscribers' login names, and the other (called NIC proxy 2 in this documentation) submits subscribers' IP addresses.

When NIC proxy 1 sends a login name to the host DemoHost, the following sequence of events occurs:

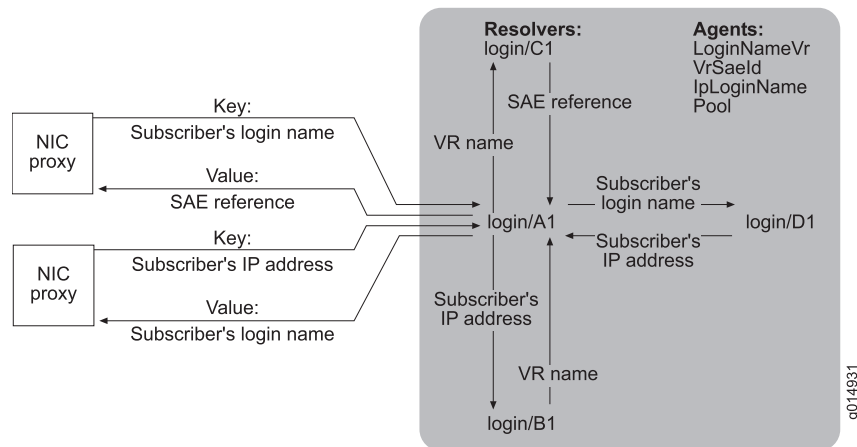
1. The host passes the login name to resolver A1.
2. Resolver A1 obtains a domain name for the login name.
3. Resolver A1 forwards the login name and the domain to resolver B1.
4. Resolver B1 obtains a VR name for the login name and returns the VR name to resolver A1.
5. Resolver A1 forwards the VR name to resolver C1.
6. Resolver C1 obtains an SAE reference for the VR and returns the SAE reference to resolver A1.
7. Resolver A1 returns the SAE reference to its host.
8. The host returns the SAE reference to the NIC proxy.

When NIC proxy 2 sends a subscriber's IP address to host DemoHost, the following sequence of events occurs.

1. The host passes the IP address to resolver A1.
2. Resolver A1 obtains an IP pool for the IP address.
3. Resolver A1 forwards the IP address and the IP pool to resolver D1.
4. Resolver D1 obtains a login name for the IP address and returns the login name to resolver A1.
5. Resolver A1 passes the login name to its host.
6. The host returns the login name to the NIC proxy.

Figure 37 on page 205 illustrates the interactions of the NIC components for this realm.

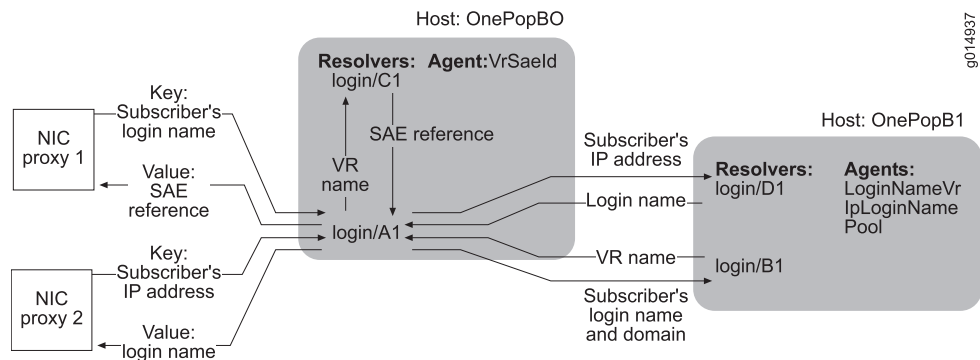
Figure 37: OnePopLogin Centralized Configuration



## Distributed Configuration

In this configuration, the agents and resolvers are distributed among several hosts. When the NIC proxy sends a subscriber's IP address to the host OnePopBO, the resolvers execute the same actions as they do in the centralized configuration. Figure 38 on page 205 illustrates the interactions of the NIC components for this realm.

Figure 38: OnePopLogin Distributed Configuration

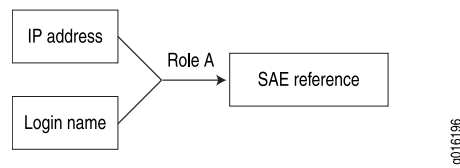


- Related Topics**
- Overview of NIC Configuration Scenarios on page 187
  - Configuring a NIC Scenario (SRC CLI) on page 134

## OnePopLoginPull Scenario

The OnePopLoginPull configuration scenario provides a simple NIC resolution from a subscriber login name or IP address to an SAE reference.

Figure 39 on page 206 shows the resolution graph for this scenario.

**Figure 39: OnePopLoginPull Distributed Configuration**

In the OnePopLoginPull scenario, SAE client agents read entries under *o=umc*, *o=servers*, *o=sspadminurls* in the Juniper Networks database to determine which SAEs are active. They also periodically check if other SAEs have become active. The SAE external interface for the active SAEs determines which SAE has a user session for the subscriber identified either by login identifier or IP identifier.

The OnePopLoginPull scenario includes the following SAE client agents:

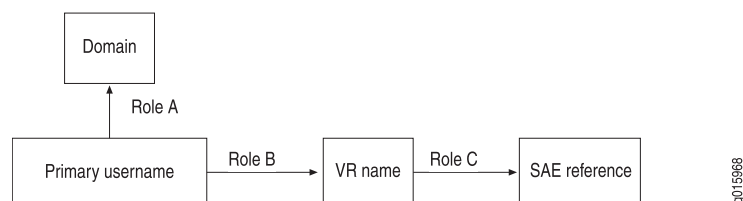
- LoginSaeld
- IpSaeld

- Related Topics**
- Overview of NIC Configuration Scenarios on page 187
  - Configuring a NIC Scenario (SRC CLI) on page 134

## OnePopPrimaryUser

The OnePopPrimaryUser scenario is similar to one of the resolutions in the OnePopLogin scenario. In the OnePopPrimaryUser scenario, subscriber primary username, as identified by the PA\_PRIMARY\_USER\_NAME attribute, is resolved to a reference for a managing SAE. The realm for this configuration accommodates a situation in which a NIC proxy provides a primary username.

Figure 40 on page 206 show the resolution graph for this realm.

**Figure 40: Resolution Processes for primary\_user Realm**

The following agents interact with resolvers in this realm:

- Directory agent VrSaeld collects and publishes information about virtual routers and the mappings between virtual routers and SAEs.
- SAE plug-in agent UserNameVr collects and publishes information about the mappings of subscriber primary usernames to VR names.

The OnePopPrimaryUser scenario provides two host configurations: a centralized configuration and a distributed configuration.

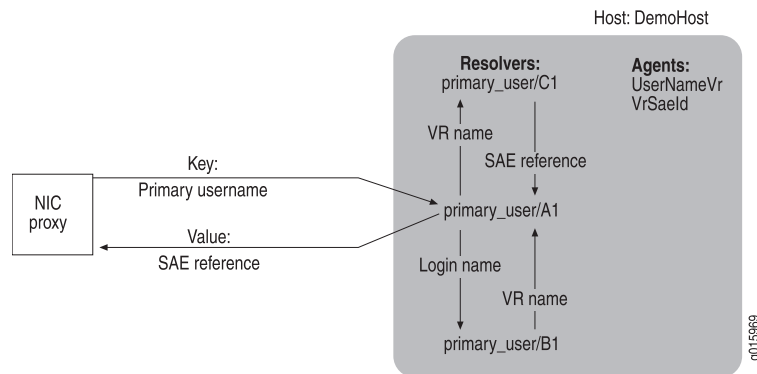
## Centralized Configuration

In this configuration, a single host called DemoHost supports all agents and resolvers. When a NIC proxy send a subscriber's primary username to host Demo Host, the following sequence of events occurs:

1. The host passes the primary username to resolver A1.
2. (Optional) Resolver A1 resolves the primary username to its domain.
3. Resolver A1 forwards the primary username to resolver B1.
4. Resolver B1 obtains the name of the VR associated with the subscriber's primary username and returns the VR to resolver A1.
5. Resolver A1 forwards the VR to resolver C1.
6. Resolver C1 obtains the SAE reference for the SAE managing the VR and returns the SAE reference to resolver A1.
7. Resolver A1 returns the SAE reference to the host.
8. The host returns the SAE reference to the NIC proxy.

Figure 41 on page 207 illustrates the interactions of the NIC components for this realm.

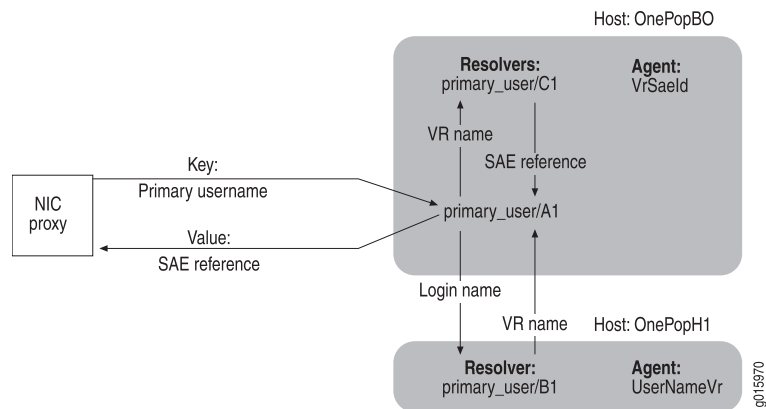
**Figure 41: OnePopPrimaryUser Centralized Configuration**



## Distributed Configuration

In this configuration, the agents and resolvers are distributed among two hosts. When a NIC proxy sends a subscriber's primary username to the host OnePopBO, the resolvers execute the same actions as they do in the centralized configuration. Figure 42 on page 208 illustrates the interactions of the NIC components for this realm.

Figure 42: OnePopPrimaryUser Distributed Configuration



- Related Topics**
- Overview of NIC Configuration Scenarios on page 187
  - Configuring a NIC Scenario (SRC CLI) on page 134

## OnePopDnSharedIp Scenario

The OnePopDnSharedIp scenario illustrates how to configure SAE plug-in agents that have state synchronization enabled to support an SAE plug-in that uses state synchronization. This scenario uses the same centralized and distributed configurations of hosts as the OnePop scenario.

Two realms are configured:

- Shared IP

The resolution process is identical to that for the OnePop scenario.

- DN realm

This realm uses essentially the same resolution process as the MultiPop DN realm. However, some of the constraints differ.

This realm also uses the same agents as the MultiPop DN realm. The names of agents and resolvers are essentially the same as those in the MultiPop configuration, although they do not include a POP identifier. Figure 43 on page 209 illustrates the centralized configuration, and Figure 44 on page 211 illustrates the distributed configuration for the DN realms.

The configuration for the two realms is similar to the configuration for the shared IP and DN realms in the OnePopAllRealms scenario. .

The OnePopAllRealms illustrates SAE plug-in agents configured to use SAE plug-in redundancy rather than SAE plug-in agents.

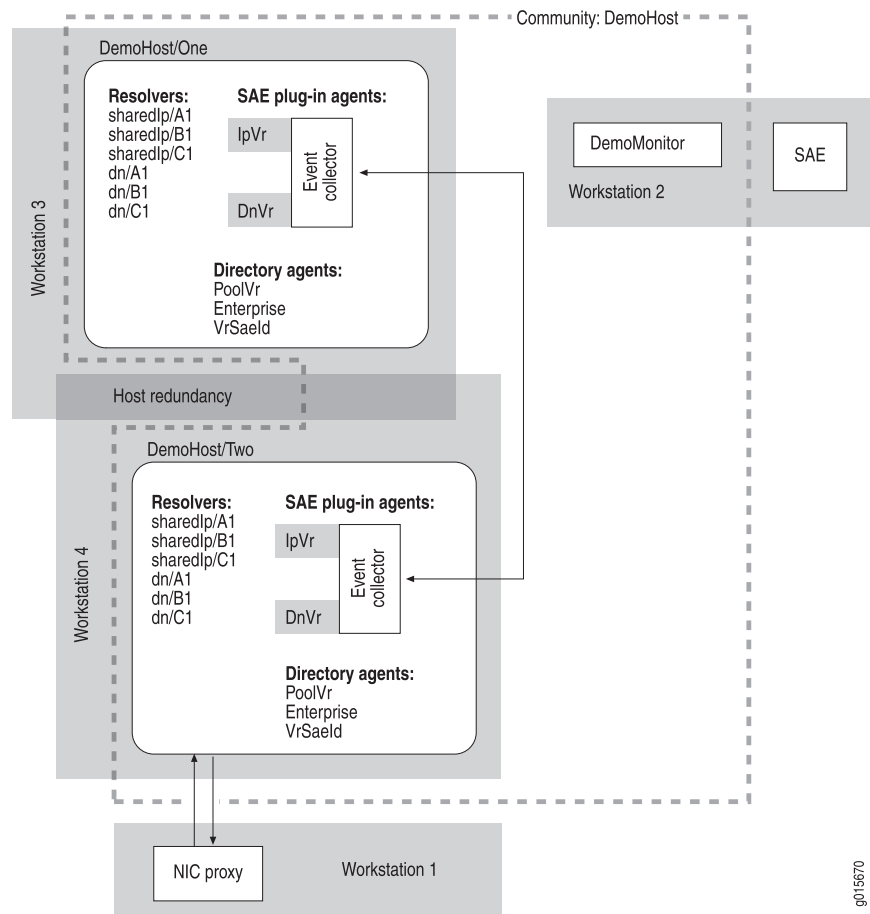


## Centralized Configuration

Figure 43 on page 209 shows the centralized configuration for the scenario. Host DemoHost supports all resolvers and agents. The two SAE plug-in agents, IpVr and DnVr, share an event collector. Both plug-in agents have state synchronization enabled.

DemoHost is also configured for redundancy. Its redundant hosts (DemoHost/One and DemoHost/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host. The redundant hosts form a community called DemoHost with the monitor DemoMonitor, which tracks them.

**Figure 43: OnePopDnSharedIp Realms Centralized Configuration**



## Distributed Configuration

Figure 44 on page 211 shows the distributed configuration from the scenario. Host OnePopBO supports two resolvers for each realm and a directory agent that is used by different realms. Host OnePopH1 supports one resolver for each realm and agents that are used by different realms.

Both hosts also have a redundant configuration. The redundant hosts for OnePopBO (OnePopBO/One and OnePopBO/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host.

The redundant hosts for OnePopH1 (OnePopH1/One and OnePopH1/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host.

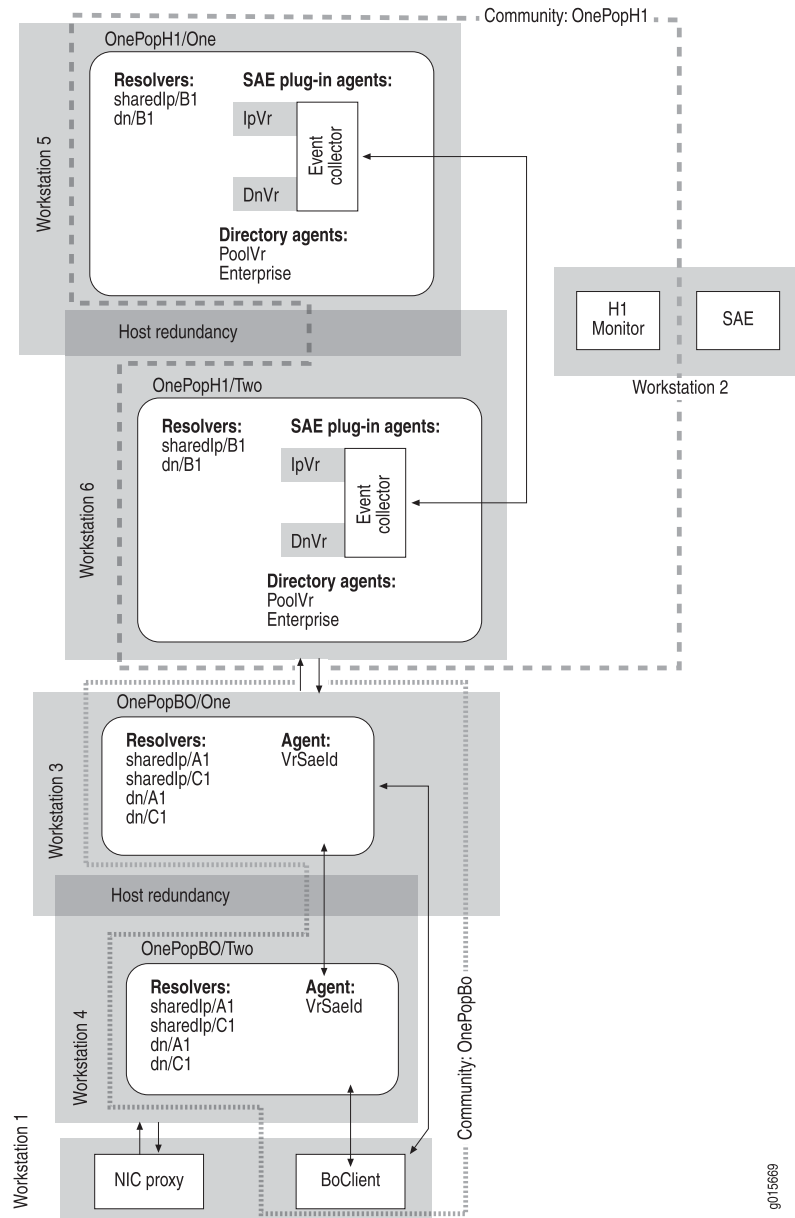
However, host OnePopH1 also supports two SAE plug-in agents, IpVr and DnVr, which share an event collector. These agents have state synchronization enabled.

The redundant hosts OnePopBO/One and OnePopBO/Two are members of a community called OnePopBO. This community supports the monitor, BoClient, which is installed on the machine that supports the NIC proxy. BoClient tracks the connections between the redundant hosts OnePopBO/One and OnePopBO/Two from the point of view of the NIC client (NIC proxy).

Similarly, the redundant hosts OnePopH1/One and OnePopH1/Two are members of a community called OnePopH1. This community has one monitor, H1Monitor, which is located on the same machine as the SAE and tracks the connections among the redundant hosts in the same community, their primary host, and the other hosts in the configuration.

H1Monitor comprises the monitor process OnePop, which is installed on the same machine as the SAE. BoClient comprises the monitor process OnePopClient, which is installed on the same machine as the NIC proxy.

Figure 44: OnePopDnSharedIp Realms Distributed Configuration



- Related Topics**
- Overview of NIC Configuration Scenarios on page 187
  - Configuring a NIC Scenario (SRC CLI) on page 134
  - OnePop Scenario on page 188
  - MultiPop Scenario on page 216

## OnePopAllRealms Scenario

---

The main purpose of the OnePopAllRealms scenario is to illustrate how to configure redundancy. This scenario uses the same centralized and distributed configurations of hosts as the OnePop scenario.

Three realms are configured:

- IP realm

This realm uses essentially the same resolution process as the IP realm for the OnePop scenario. However, some of the constraints differ.

- Shared IP

The resolution process is identical to that for the OnePopShared scenario.

- DN realm

This realm uses essentially the same resolution process as the MultiPop DN realm. However, some of the constraints differ.

This realm also uses the same agents as the MultiPop DN realm. The names of agents and resolvers are essentially the same as those in the MultiPop configuration, although they do not include a POP identifier. By reviewing the scenario, Figure 45 on page 213 and Figure 46 on page 215, you can determine exact pictures of the DN realms for the centralized and distributed configurations.

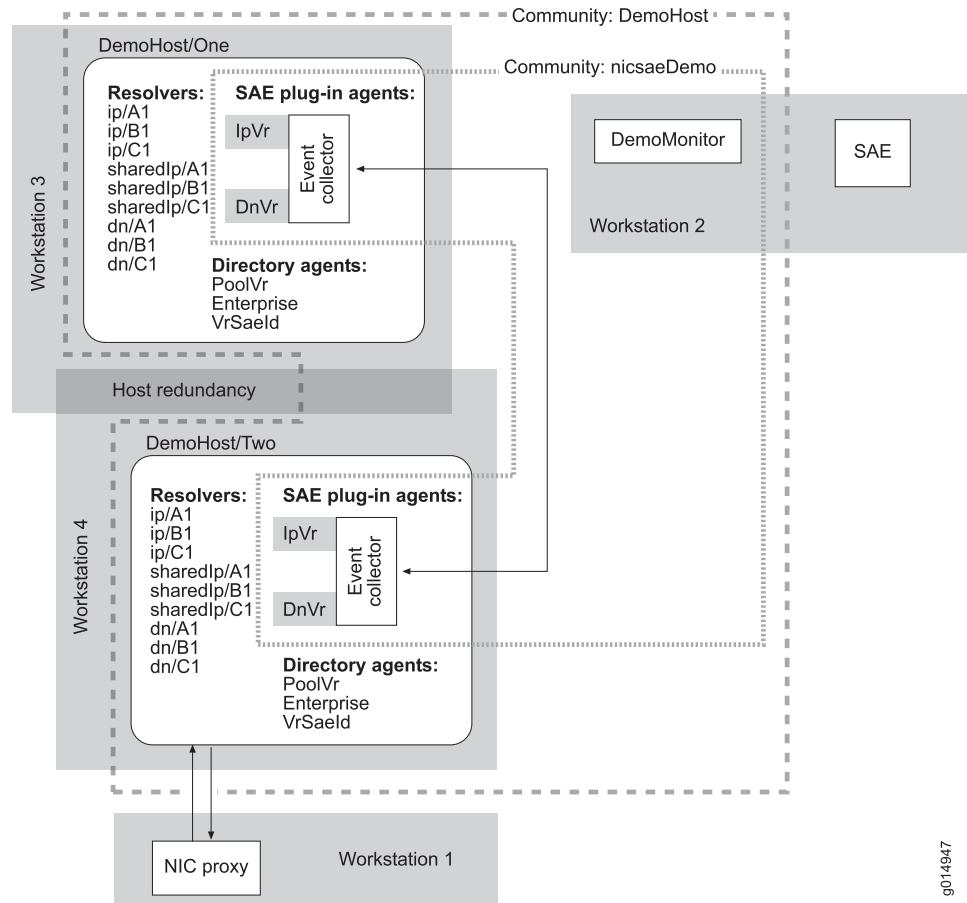
Figure 45 on page 213 shows the centralized configuration for the scenario. Host DemoHost supports all resolvers and agents. However, because host DemoHost is configured for redundancy, its redundant hosts (DemoHost/One and DemoHost/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host.

The parent host DemoHost also supports two SAE plug-in agents, IpVr and DnVr, which share an event collector. Each SAE plug-in agent has a redundant agent called Demo; these redundant agents also share an event collector. The redundant agents and their shared event collector are assigned to both redundant hosts DemoHost/One and DemoHost/Two.

The redundant agents form a community called nicsaeDemo with the monitor DemoMonitor, which tracks them. The redundant agents are identified in the community by the names DemoHost/One and DemoHost/Two; these names specify their hosts and provide unique identifiers for the redundant agents.

The redundant hosts form a community called DemoHost with the monitor DemoMonitor, which tracks them.

Figure 45: OnePopAllRealms Centralized Configuration



g014947

Figure 46 on page 215 shows the distributed configuration for the scenario. Host OnePopBO supports two resolvers for each realm and a directory agent that is used by different realms. However, because host OnePopBO is configured for redundancy, its redundant hosts (OnePopBO/One and OnePopBO/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host.

Host OnePopH1 supports one resolver for each realm and agents that are used by different realms. Host OnePopH1 is also configured for redundancy, and its redundant hosts (OnePopH1/One and OnePopH1/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host.

However, host OnePopH1 also supports two SAE plug-in agents, IpVr and DnVr, which share an event collector. Each SAE plug-in agent has a redundant agent called onePop; these redundant agents also share an event collector. The redundant agents and their shared event collector are assigned to redundant hosts OnePopH1/One and OnePopH1/Two.

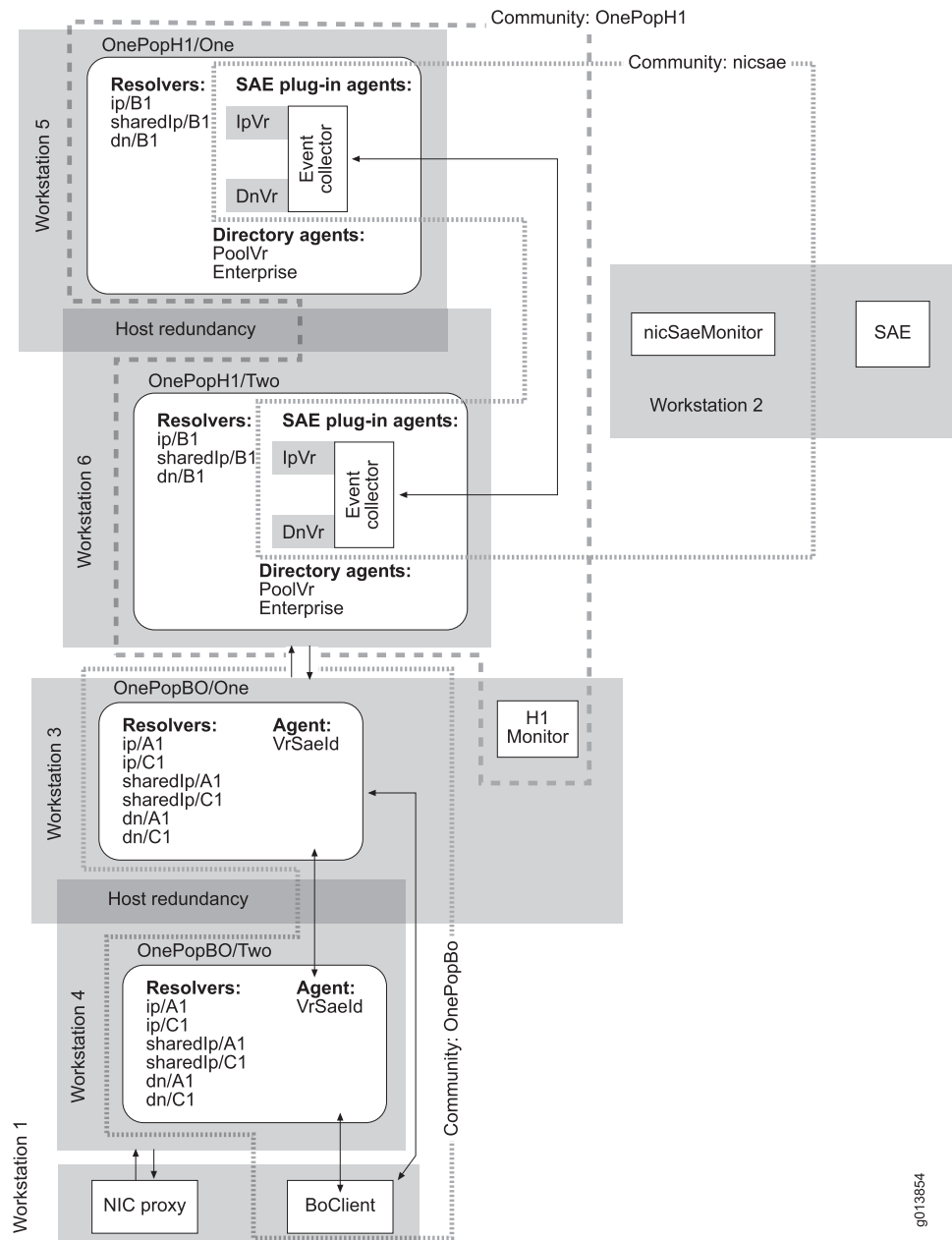
The redundant agents form a community called nicsae with monitor nicSaeMonitor, which tracks them. The redundant agents are identified in the community by the names OnePopH1/One and OnePopH1/Two; these names specify their hosts and provide unique identifiers for the redundant agents.

The redundant hosts OnePopBO/One and OnePopBO/Two are members of a community called OnePopBO. This community supports the monitor, BoClient, which is installed on the machine that supports the NIC proxy. BoClient tracks the connections between the redundant hosts OnePopBO/One and OnePopBO/Two from the point of view of the NIC client (NIC proxy).

Similarly, the redundant hosts OnePopH1/One and OnePopH1/Two are members of a community called OnePopH1. This community has one monitor, H1Monitor, which is located on the same machine as the SAE and tracks the connections among the redundant hosts in the same community, their primary host, and the other hosts in the configuration.

H1Monitor and nicSaeMonitor are part of the monitor process OnePop, which is also installed on the same machine as the SAE. BoClient is part of the monitor process OnePopClient, which is installed on the same machine as the NIC proxy.

Figure 46: OnePopAllRealms Distributed Configuration



g013854

- Related Topics**
- Overview of NIC Configuration Scenarios on page 187
  - Configuring a NIC Scenario (SRC CLI) on page 134
  - OnePop Scenario on page 188
  - OnePopSharedIp Scenario on page 194
  - MultiPop Scenario on page 216

## MultiPop Scenario

---

The MultiPop scenario illustrates a configuration that involves two POPs: Montreal and Ottawa. This configuration does not provide redundancy. The NIC proxy communicates with the back office host (BackOffice), which in turn communicates with the POP hosts (MontrealHost and OttawaHost). Hosts MontrealHost and OttawaHost support equivalent hosts and agents and manage resolutions in the same way.

When host BackOffice receives a data key from the NIC proxy, the following sequence of events occurs:

1. Host BackOffice forwards requests as follows:
  - If the request is for the Montreal POP, host BackOffice forwards the request to POP host MontrealHost.
  - If the request is for the Ottawa POP, host BackOffice forwards the request to POP host OttawaHost.
2. Delegating tasks to other resolvers as necessary, the resolvers in the POP obtain data values that correspond to the data key request, and return them.
3. The POP host returns the data values to host BackOffice, which returns the value to the NIC proxy.

The scenario shows three realms for this configuration:

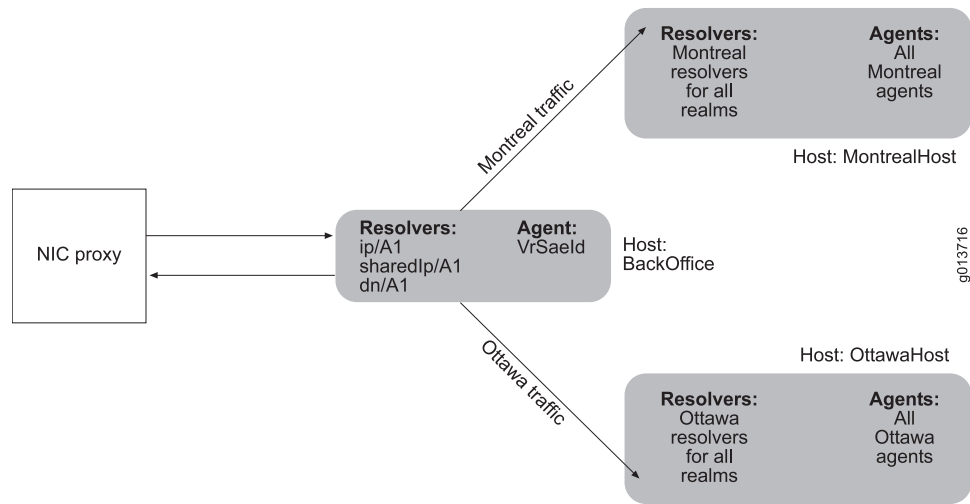
- IP
- Shared IP
- DN

Each realm provides a different type of resolution. The following sections provide information about these realms.

Figure 47 on page 217 illustrates this configuration.



Figure 47: MultiPop Configuration



## IP Realm

This realm accommodates the situation in which IP address pools are configured locally on each VR. The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value. This realm uses essentially the same resolution process as the ip realm for the OnePop scenario (see "OnePop Scenario" on page 188). However, some of the constraints differ.

The following agents interact with the resolvers in this realm:

- Directory agents montrealPoolVr and ottawaPoolVr collect and publish information that maps IP address pools to VRs. Each agent publishes only the information that is relevant to its POP. You achieve selective publishing by relating an Ottawa scope to the VRs in the Ottawa POP and a Montreal scope to the VRs in the Montreal POP and defining a search filter for the agents to load only the VRs in its POP.
- Directory agent VrSaeld in the back office collects and publishes information that maps VRs to SAEs for both POPs.

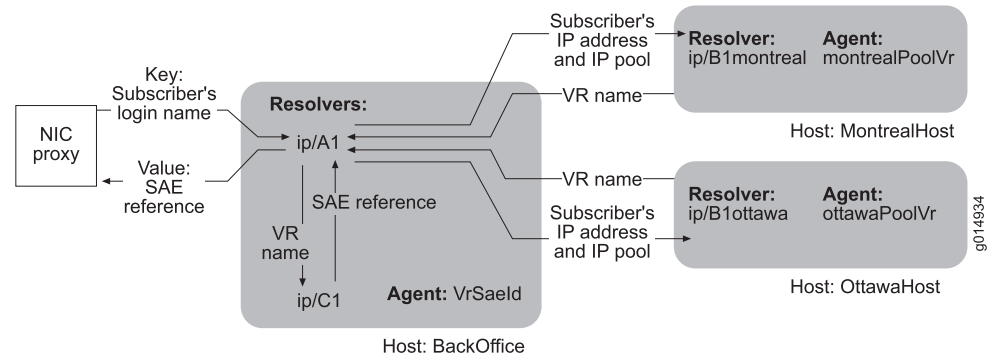
When the NIC proxy sends a subscriber's IP address to host BackOffice, the following sequence of events occurs:

1. Host BackOffice passes the IP address to resolver ip/A1.
2. Resolver ip/A1 obtains an IP pool for the IP address.
3. Resolver ip/A1, based on the value of the IpPool, forwards the request to ip/B1montreal or ip/B1ottawa.
4. Resolver ip/B1montreal or resolver ip/B1ottawa obtains a VR name for this IP pool and returns the VR name to resolver ip/A1.
5. Resolver ip/A1 forwards the VR name to resolver ip/C1.
6. Resolver ip/C1 obtains the SAE identity for this VR and returns the value to resolver ip/A1.

7. Resolver ip/A1 returns the SAE reference to its host.
8. Host BackOffice returns the SAE reference to the NIC proxy.

Figure 48 on page 218 illustrates the interactions of the NIC components for this realm.

**Figure 48: iP Realm for MultiPop Configuration**



## Shared IP Realm

This realm accommodates the situation in which IP address pools are shared by VRs in the same POP. The realm takes a subscriber's IP address as the key and returns the corresponding SAE as the value. To see the resolution graph for this realm, see "OnePop Scenario" on page 188.

The following agents interact with resolvers in this realm:

- Directory agents montrealPoolVr and ottawaPoolVr collect and publish information about the mappings of IP address pools to VRs. Each agent publishes only the information that is relevant to its POP.
- SAE plug-in agents montrealIpVr and ottawaIpVr collect and publish information about the mappings of subscriber IP addresses to VRs. Each agent publishes only the information that is relevant to its POP.
- Directory agent VrSaeld in the back office collects and publishes information about the mappings of VRs to SAEs for both POPs.

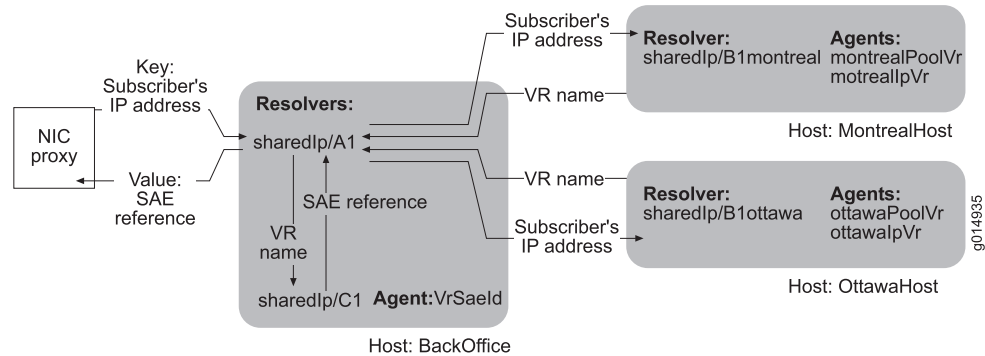
When the NIC proxy sends a subscriber's IP address to host BackOffice, the following sequence of events occurs:

1. Host BackOffice passes the IP address to resolver sharedIp/A1.
2. Resolver sharedIp/A1 obtains an IP pool for the IP address.
3. Resolver sharedIp/A1, based on the value of the IP pool, forwards the request to sharedIp/B1montreal or sharedIp/B1ottawa.
4. Resolver sharedIp/B1montreal or resolver sharedIp/B1ottawa obtains a VR name for this IP address and returns the VR name to resolver sharedIp/A1.
5. Resolver sharedIp/A1 forwards the VR name to resolver sharedIp/C1.

6. Resolver sharedIp/C1 obtains the SAE identity for this VR and returns the value to resolver sharedIp/A1.
7. Resolver sharedIp/A1 passes the SAE reference to its host.
8. Host BackOffice returns the SAE reference to the NIC proxy.

Figure 49 on page 219 illustrates the interactions of the NIC components for this realm.

**Figure 49: sharedIP Realm for MultiPop Configuration**

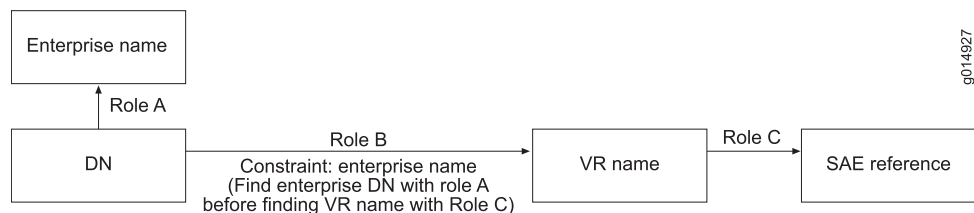


## DN Realm

The DN realm takes the DN of an access subscriber (an access DN) as the key and returns the corresponding SAE as the value. Figure 50 on page 219 shows the resolution process for this realm.

Figure 50 on page 219 shows the resolution graph for this realm.

**Figure 50: Resolution Graph for MultiPOP dn Realm**



The following agents interact with resolvers in this realm:

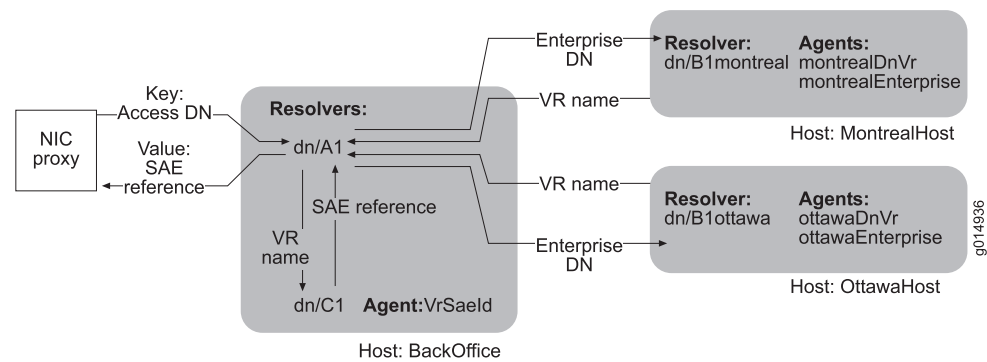
- Directory agents **ottawaEnterprise** and **montrealEnterprise** collect and publish information about the DNs of enterprise subscribers (enterprise DNs). Each agent publishes only the information that is relevant to its POP. You achieve selective publishing by relating an Ottawa service scope to the enterprises in the Ottawa POP and a Montreal service scope to the enterprises in the Montreal POP and defining a search filter for the agents to load only the enterprises in its POP.
- SAE plug-in agents **montrealDnVr** and **ottawaDnVr** collect and publish information about the mappings of access DNs to VRs. Each agent publishes only the information that is relevant to its POP.
- Directory agent **VrSaeld** collects and publishes information about the mappings of VRs to SAEs for both POPs.

When the NIC proxy sends an access DN to host BackOffice, the following sequence of events occurs:

1. Host BackOffice passes the access DN to resolver dn/A1.
2. Resolver dn/A1 obtains an enterprise DN for the access DN.
3. Resolver dn/A1, based on the value of the enterprise DN, forwards the request to dn/B1montreal or dn/B1ottawa.
4. Resolver dn/B1montreal or resolver dn/B1ottawa obtains a VR name for this enterprise DN and returns the VR name to resolver dn/A1.
5. Resolver dn/A1 forwards the VR name to resolver dn/C1.
6. Resolver dn/C1 obtains the SAE reference for this VR and returns the value to resolver dn/A1.
7. Resolver dn/A1 passes the SAE reference to its host.
8. Host BackOffice returns the SAE reference to the NIC proxy.

Figure 51 on page 220 illustrates the interactions of the NIC components for this realm.

**Figure 51: dn Realm for MultiPop Configuration**



- Related Topics**
- Overview of NIC Configuration Scenarios on page 187
  - Configuring a NIC Scenario (SRC CLI) on page 134

## PART 5

# Providing Admission Control with SRC ACP

- Overview of Providing Admission Control with SRC ACP on page 223
- Configuring Admission Control (SRC CLI) on page 233
- Configuring Congestion Point Classification (SRC CLI) on page 269
- Managing SRC ACP (SRC CLI) on page 279
- Monitoring Admission Control (SRC CLI) on page 283
- Monitoring Admission Control (C-Web Interface) on page 293



# Overview of Providing Admission Control with SRC ACP

- Overview of SRC ACP on page 223
- Deriving Congestion Points Automatically on page 225
- Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 227
- Use of Multiple SRC ACPs on page 228
- Interactions Between SRC ACP and Other Components on page 228
- Redundancy and State Synchronization on page 230
- Fault Recovery on page 231
- Creating an Application to Update Information for SRC ACP on page 231

## Overview of SRC ACP

---

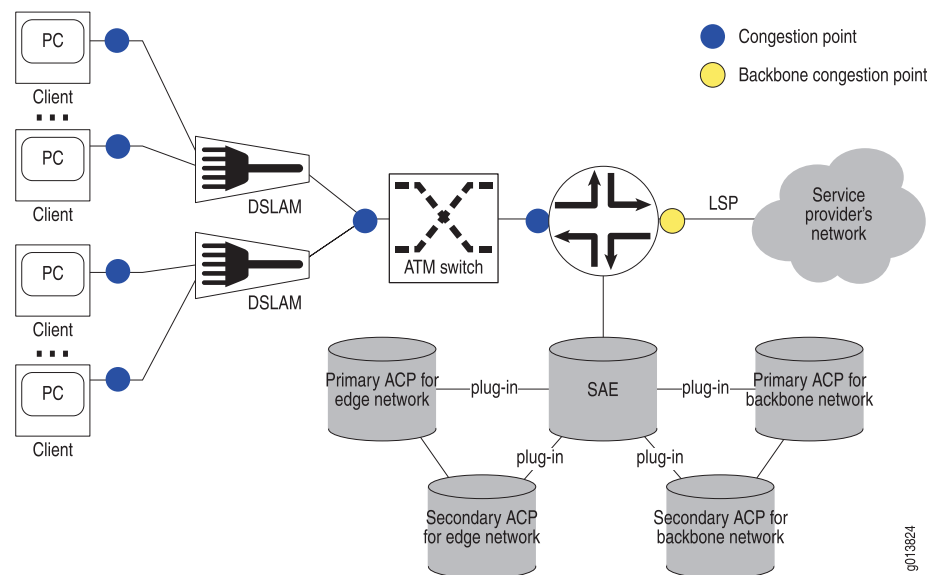
SRC ACP is an external plug-in for the SAE. SRC ACP authorizes and tracks subscribers' use of network resources associated with services that the SRC software manages. Service providers can implement SRC ACP configurations for both residential and enterprise subscribers. Consequently, both JUNOSe routers and JUNOS routing platforms are compatible with SRC ACP. References to virtual routers (VRs) in this documentation refer to an actual VR on a JUNOSe router or the single VR called default that the SRC software associates with each JUNOS routing platform.

SRC ACP operates in two separate regions of the SRC network: the *edge* network and the *backbone* network. The edge network is the layer 2 access network through which subscribers connect to the router. The backbone network is the region between the router and the service provider's network.

Congestion often occurs in the network at points where connections are aggregated. SRC ACP monitors congestion points at interfaces between devices in the edge network. In the backbone network, SRC ACP monitors one congestion point, a point-to-point label-switched path (LSP) between the router and the service provider's network.

Figure 52 on page 224 shows a typical network topology.

Figure 52: Position of SRC ACP in Network



In the edge network, SRC ACP performs the following procedures to determine whether there are sufficient resources to activate a service:

- Tracks active services for each subscriber and the guaranteed traffic rate (bandwidth) at the congestion points associated with a subscriber.
- Tracks the rate of traffic between the subscriber and the network (upstream bandwidth) and the rate of traffic between the network and subscriber (downstream bandwidth).
- Monitors new requests for activation of services.
- Compares the resources required for the new services with the resources available for the subscriber and the congestion points.
- Activates the service if sufficient resources are available, and prevents activation of the service if sufficient resources are not available.

In the backbone network, SRC ACP performs the following procedures to determine whether there are sufficient resources to activate a service:

- Tracks the guaranteed traffic rate for a service at the congestion point.
- Tracks the actual traffic rate for the service at the congestion point.
- Monitors new requests for activation of services.
- Compares the resources required for the new services with the resources available at the congestion point.
- Activates the service if sufficient resources are available, and prevents activation of the service if sufficient resources are not available.

Typically, network administrators use their own network management applications and external applications to provide data for SRC ACP. SRC ACP first obtains updates from external applications through its remote CORBA interface, and then obtains updates



from the directory by means of LDAP. For information about developing external applications that send data to SRC ACP, see “Creating an Application to Update Information for SRC ACP” on page 231. SRC ACP does not interact directly with the network to assess the capacity of a congestion point or actual use of network resources.

In the backbone network, SRC ACP can also execute applications defined in the action congestion point. Some applications require real-time congestion point status. If SRC ACP must provide real-time congestion point status to the application, state synchronization must be enabled to handle interface tracking events so that the congestion points are updated properly.

- Related Topics**
- Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 227
  - Use of Multiple SRC ACPs on page 228
  - Interactions Between SRC ACP and Other Components on page 228
  - Configuring SRC ACP (SRC CLI) on page 236

---

## Deriving Congestion Points Automatically

SRC ACP can derive some congestion points automatically. Depending on your network configuration and requirements, however, you may need to enter congestion points manually. This topic describes the conditions and requirements for SRC ACP to derive congestion points automatically.

### Deriving Edge Congestion Points

For SRC ACP to derive edge congestion points, subscribers must always connect through the same interface on the router. In addition, SRC ACP requires one of the following conditions to derive edge congestion points if you are not using a congestion point profile:

- Access to subscriber profiles that define bandwidth values and a list of the distinguished names (DNs) of congestion points between the subscriber and the router.
- An ATM access network between the subscriber and the router for which all the traffic coming from one DSLAM travels on a single virtual path. In this case, SRC ACP automatically derives three congestion points through the network access server (NAS) port ID. Table 13 on page 226 shows the edge congestion points and the corresponding locations in the directory.

For information about the NAS port ID, see Using Flexible RADIUS Packet Definitions.

SRC ACP does not use bandwidth statistics from subscriber profiles when it derives congestion points, because the congestion points already use that data.

Table 13: Congestion Points Derived Through NAS Port ID

| Congestion Points            | Location of Object in Directory                                                                                                                                                                                                      |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Physical interface on router | interfaceName=ATM<slot>/<port>, <i>orderedCimKeys=&lt;routerName&gt;, o=AdmissionControl, o=umc</i><br><br><slot>—Number of port on router<br><br><port>—Number of port on router<br><br><routerName>—Hostname configured for router |
| ATM virtual path             | interfaceName=ATM<slot>/<port>:<vpi><br><i>orderedCimKeys=&lt;routerName&gt;, o=AdmissionControl, o=umc</i><br><br><vpi>—Number of virtual path on router                                                                            |
| ATM virtual connection       | interfaceName=ATM<slot>/<port>:<vpi>.<vci><br><i>orderedCimKeys=&lt;routerName&gt;, o=AdmissionControl, o=umc</i><br><br><vci>—Number of virtual connection on router                                                                |

### Deriving Congestion Points from a Profile

If you configure a congestion point profile, SRC ACP can automatically derive congestion points for cases in which:

- There is no subscriber profile.
- The congestion points can be derived from information provided by the access interface on B-RAS. For example, in an ATM or VLAN connection, you can derive congestion points representing physical interfaces and intermediate switches based on the NAS port ID reported by B-RAS.

When SRC ACP receives notification to start subscriber tracking and to load congestion points for a subscriber, it runs a congestion point classification and accesses the configured congestion point profile. Congestion point classification uses the same classification engine as subscriber and interface classification in the SAE.

For this feature to operate correctly, you create a congestion point profile that automatically performs congestion point classification.

### Deriving Backbone Congestion Points

SRC ACP can automatically derive backbone congestion points if you specify the setting <-vrName->/<-serviceName-> for the congestion point associated with a service. When the SRC ACP starts operating, it will substitute the name of the VR and the service name from the activation request.

For example, you can specify the setting <-vrName->/<-serviceName-> for the congestion point associated with a service called News. Then, when a subscriber who connects to the network through a VR called boston requests activation of this service, SRC ACP receives the request and proceeds as follows:

1. SRC ACP reads the congestion point specification, `<-vrName->/<-serviceName->`, from the congestion point defined for the service `News`.
2. SRC ACP substitutes the actual information, `boston/News`, in the variables.
3. SRC ACP uses this information to generate the DN `cn=News, cn=boston, o=CongestionPoints, o=umc`.
4. SRC ACP uses this DN to obtain from the directory the network interface, which defines the location of the congestion point, for this DN.

For this feature to operate correctly, you must configure the DN for each combination of VR and service to point to an actual network interface.

In cases where the combination of VR and service name do not uniquely identify the backbone congestion point, you can use backbone congestion point expressions and Python scripts to classify the backbone congestion point. Python scripts are executed when evaluating the congestion point expression. The format of the backbone congestion point expression is similar to the expression used in the congestion point profile. You can embed Python expressions, such as service plug-in attributes, in the congestion point expression. As a result, you can derive multiple backbone congestion points from a single service session.

For example, you can have a video-on-demand service that uses multiple video servers. One label-switched path (LSP) with the same parameters is created for each link between a video server and an access router. SRC ACP uses the network interface configuration information to generate the DN `interfaceName=<NetworkInterface>`, `orderedCimKeys=<NetworkDevice>`, `o=AdmissionControl`, `o=umc` as a template for the congestion point. When receiving a service request, the server activates the service for the subscriber on the appropriate congestion point. The backbone congestion point corresponds to the evaluation of the backbone congestion point expression.

- Related Topics**
- Defining a Congestion Point Profile (SRC CLI) on page 276
  - Configuring SRC ACP (SRC CLI) on page 236
  - Overview of Congestion Point Classification on page 269
  - Viewing Congestion Point Information by DN (SRC CLI) on page 290
  - Viewing Congestion Point Information by Name (SRC CLI) on page 290

---

## Allocating Bandwidth to Applications Not Controlled by SRC ACP

If you control the bandwidth of some applications by means of SRC ACP, you can accommodate the applications that are not controlled by SRC ACP by assigning *background* bandwidths for the edge congestion points. The background bandwidth is the total bandwidth allocated to the applications for which bandwidth is not controlled by SRC ACP.

Because the total background bandwidth is unlikely to be used at a particular time, you can also specify a tuning factor that provides an estimation of the fraction of the

background bandwidth that will be used. You can configure multiple values for the background bandwidth with corresponding tuning factors.

- Related Topics**
- Use of Multiple SRC ACPs on page 228
  - Interactions Between SRC ACP and Other Components on page 228
  - Overview of SRC ACP on page 223
  - Configuring SRC ACP to Manage the Edge Network (SRC CLI) on page 255

---

## Use of Multiple SRC ACPs

An SRC ACP can support one or more SAEs. Similarly, multiple SRC ACPs can support one SAE; for example, if an SAE is managing multiple VRs, you may have an SRC ACP for each VR. However, only one SRC ACP can manage a particular congestion point.

- Related Topics**
- Overview of SRC ACP on page 223
  - Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 227
  - Interactions Between SRC ACP and Other Components on page 228
  - Configuring the SAE for SRC ACP (SRC CLI) on page 241
  - Configuring SRC ACP (SRC CLI) on page 236
  - Configuring SRC ACP Properties (SRC CLI) on page 243

---

## Interactions Between SRC ACP and Other Components

This topic describes how SRC ACP interacts with other components to track data.

1. (Edge and dual mode only) When a subscriber connects to the router, SRC ACP loads the subscriber profile from the directory. If the subscriber profile contains provisioned and actual traffic rates for the subscriber's interface and the set of congestion points between the subscriber and the router, SRC ACP caches the information while the subscriber is connected to the router. SRC ACP automatically updates the subscriber's actual upstream and downstream rates if the subscriber profile changes in the directory.
2. (Backbone mode only) When a subscriber activates a service, SRC ACP loads the network interfaces defined in the service and caches the information.
3. (Optional) SRC ACP obtains through its remote CORBA interface data from external applications about subscribers and congestion points. If a congestion point is unavailable, SRC ACP denies service activation requests on the associated network interface until the interface is available again.
4. If SRC ACP does not receive data from an external application, SRC ACP loads data about congestion points from the directory. For each congestion point the following data is retrieved:
  - Provisioned bandwidth

- Background bandwidths (if used for edge congestion points)

SRC ACP caches this information and automatically updates the cache when the information changes in the directory.

5. (Edge and dual modes) If SRC ACP does not receive data from an external application, SRC ACP loads a subscriber's provisioned or actual bandwidth from the subscriber profile. If the actual bandwidth is available, SRC ACP ignores the provisioned bandwidth.

SRC ACP caches this information and automatically updates the cache when the information changes in the directory.

6. (Backbone and dual modes only) Using a hosted plug-in, the SAE monitors the states of router interfaces associated with backbone congestion points. The SAE sends relevant data to SRC ACP through the SRC ACP's remote interface.
7. When the subscriber requests activation of a service subscription (either through the SAE core API or automatically for activate-on-login services), the SAE notifies SRC ACP to authorize and track the service usage.
  - a. The SAE sends the requested bandwidth to SRC ACP.
  - b. SRC ACP authorizes or denies service activation.

If SRC ACP authorizes the service activation, the SAE activates the service and sends a tracking event to SRC ACP. SRC ACP updates the current bandwidth for all congestion points with the requested bandwidth.

If SRC ACP authorizes the service activation with state synchronization enabled, SRC ACP reserves the requested bandwidth on all congestion points until the reservation expires. You can specify the reservation timeout value when configuring SRC ACP operation.

- For each congestion point, SRC ACP verifies whether:

$$(\text{current bw} + \text{reserved bw} + \text{requested bw}) > [\text{provisioned bw} - (\text{background bw} \times \text{tuning factor})]$$

If the desired bandwidth exceeds the allocated bandwidth, SRC ACP denies service activation.

- When SRC ACP receives a service start tracking event, the requested bandwidth is committed. That is, for each congestion point, the requested bandwidth reservation is removed and the requested bandwidth is added to the current bandwidth.
- When the bandwidth reservation expires, the reserved bandwidth is released.

If SRC ACP does not authorize the service activation, the SAE delivers a message detailing the reason to the originator of the activation request.

SRC ACP distinguishes between bandwidth exceeded on the subscriber interface (first congestion point) and bandwidth exceeded on a network interface by sending two different messages back to the SAE. In the first case, the subscriber may resolve the bandwidth problem by deactivating another service.

8. When a service is deactivated (either through the SAE core API or because a session times out), SRC ACP updates the current bandwidth for all congestion points by removing the original requested bandwidth.
9. SRC ACP stores all information about subscribers, services, and congestion points in a set of files.

SRC ACP continually adds data to these files, but does not delete old data. Consequently, the sizes of the files continue to increase. SRC ACP does, however, reorganize the files when the sum of their sizes increments by a specified value. Reorganizing the files reduces their sizes. You can also reorganize the files by using the SRC CLI (see “Reorganizing the File That Contains ACP Data” on page 280 .)

**Related Topics**

- Overview of SRC ACP on page 223
- Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 227
- Use of Multiple SRC ACPs on page 228
- Configuration Statements for SRC ACP on page 233
- Configuring SRC ACP (SRC CLI) on page 236
- Configuring SRC ACP Properties (SRC CLI) on page 243

---

## Redundancy and State Synchronization

---

You can configure SRC ACP to synchronize states with the SAE.

State synchronization enables the current state to be transferred when SRC ACP has started up or lost its state. SRC ACP does not have to keep a local and persistent copy of the state. However, SRC ACP requires additional bandwidth to transfer state information that can affect performance.

You can configure SRC ACP redundancy for a region of the network by installing SRC ACP on two different hosts and connecting both SRC ACP hosts to the SAE (see Figure 52 on page 224). One SRC ACP acts as the primary application, and the other as the secondary application.



**NOTE:** Both SRC ACPs in a redundant pair must operate in the same mode. You cannot configure an SRC ACP in edge mode and an SRC ACP in backbone mode as a redundant pair.

---

To configure SRC ACP redundancy, enable redundancy. In this situation, the primary and secondary SRC ACPs are set up as a community and will communicate with each other to determine the primary SRC ACP. The primary SRC ACP registers its interoperable object reference (IOR) with the SAE so that the SAE will communicate only with the primary SRC ACP. When the primary SRC ACP becomes unavailable, the secondary SRC ACP assumes the role of the primary SRC ACP and performs state synchronization if necessary.

- Related Topics**
- Overview of SRC ACP on page 223
  - Interactions Between SRC ACP and Other Components on page 228
  - Configuration Statements for SRC ACP on page 233
  - Configuring SRC ACP (SRC CLI) on page 236
  - Configuring SRC ACP Properties (SRC CLI) on page 243

---

## Fault Recovery

If the SAE cannot reach SRC ACP, the SAE will deny all service activation requests. As soon as it reaches SRC ACP, the SAE again sends authorization requests to SRC ACP.

SRC ACP keeps the state of the congestion points in persistent storage, and if SRC ACP becomes unavailable, the service authorization can continue in the correct state. Because service activation requests are automatically denied when the SAE cannot reach SRC ACP, SRC ACP does not miss any active service sessions. The SAE will resend all service deactivation requests after SRC ACP is reachable again.

SRC ACP monitors SAE synchronization events for information about VR availability and SAE availability. If a VR reboots or an SAE becomes unavailable, SRC ACP updates the states of congestion points associated with those devices accordingly.

If the SAE becomes unavailable, the router will automatically reestablish connection to either the redundant SAE or, if a redundant SAE is not available, to the original SAE when it again becomes available. The new SAE notifies SRC ACP that the original SAE failed and specifies which subscriber and service sessions were logged during this time. SRC ACP uses this information to update its state.

- Related Topics**
- Overview of SRC ACP on page 223
  - Interactions Between SRC ACP and Other Components on page 228
  - Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 227
  - Use of Multiple SRC ACPs on page 228

---

## Creating an Application to Update Information for SRC ACP

You can develop your own application to update information about subscribers and congestion points for SRC ACP. The application can call one method to interact with SRC ACP. This method is called:

update (in RemoteUpdateType rut, in TagValueList attrs)

The method takes a property-value pair and passes the information to SRC ACP. For information about the properties and values you can pass to SRC ACP, see the file *acpPlugin.idl* in the folder *SDK/idl* in the *SDK+AppSupport+Demos+Samples.tar.gz* file on the Juniper Networks Web site at: <https://www.juniper.net/support/csc/swdist-erx/src.html>

To create an application that updates SRC ACP remotely:

1. Compile the IDL file, and generate the code in the language in which you want to write the application.
2. Write the application, and include the generated code for the IDL file.
3. Use the CORBA object reference defined in the property `ACP.syncRateAdaptor.ior` to send data from the application to SRC ACP.

For information about the interfaces, properties, and methods available in the CORBA remote API for ACP, see the documentation in the `SDK+AppSupport+Demos+Samples.tar.gz` file on the Juniper Networks Web site at: <https://www.juniper.net/support/csc/swdist-erx/src.html>. The files are in the `SDK/doc/idl/acp/html/index.html` directory.

**Related Topics**

- Overview of SRC ACP on page 223
- Interactions Between SRC ACP and Other Components on page 228
- Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 227
- Use of Multiple SRC ACPs on page 228



## Configuring Admission Control (SRC CLI)

- Configuration Statements for SRC ACP on page 233
- Configuring SRC ACP (SRC CLI) on page 236
- Creating Grouped Configurations for SRC ACP (SRC CLI) on page 236
- Configuring Local Properties for SRC ACP (SRC CLI) on page 237
- Configuring the SAE for SRC ACP (SRC CLI) on page 241
- Configuring SRC ACP Properties (SRC CLI) on page 243
- Configuring SRC ACP to Manage the Edge Network (SRC CLI) on page 255
- Configuring SRC ACP to Manage the Backbone Network (SRC CLI) on page 259

### Configuration Statements for SRC ACP

---

Use the following configuration statements to configure SRC ACP at the **[edit]** hierarchy level:

```
shared acp configuration acp-options {
 backup-directory backup-directory;
 mode (edge | backbone | dual);
 event-cache-size event-cache-size;
 overload-method overload-method;
 reservation-timeout reservation-timeout;
 congestion-point-auto-completion;
 tuning-factor tuning-factor;
 subscriber-bandwidth-exceed-message subscriber-bandwidth-exceed-message;
 network-bandwidth-exceed-message network-bandwidth-exceed-message;
 backup-database-maximum-size backup-database-maximum-size;
 remote-update-database-index-keys remote-update-database-index-keys;
 interface-tracking-filter interface-tracking-filter;
 state-sync-bulk-size state-sync-bulk-size;
}
shared acp configuration corba {
 acp-ior acp-ior;
 remote-update-ior remote-update-ior;
}
shared acp configuration ldap service-data {
 edge-congestion-point-dn edge-congestion-point-dn;
 backbone-congestion-point-dn backbone-congestion-point-dn;
 reload-congestion-points;
 congestion-points-eventing;
```

```
server-address server-address;
server-port server-port;
dn dn;
principal principal;
password password;
event-dn event-dn;
directory-eventing;
polling-interval polling-interval;
}
shared acp configuration ldap subscriber-data {
 congestion-points-eventing;
 server-address server-address;
 server-port server-port;
 dn dn;
 principal principal;
 password password;
 event-dn event-dn;
 directory-eventing;
 polling-interval polling-interval;
}
shared acp configuration logger name ...
shared acp configuration logger name file {
 filter filter;
 filename filename;
 rollover-filename rollover-filename;
 maximum-file-size maximum-file-size;
}
shared acp configuration logger name syslog {
 filter filter;
 host host;
 facility facility;
 format format;
}
shared acp configuration redundancy {
 enable-redundancy;
 local-ior local-ior;
 remote-ior remote-ior;
 ignore-user-tracking-out-of-sync;
 community-heartbeat community-heartbeat;
 community-acquire-timeout community-acquire-timeout;
 community-blackout-timeout community-blackout-timeout;
 redundant-naming-service redundant-naming-service;
}
shared acp configuration scripts-and-classification {
 script-factory-class script-factory-class;
 classification-factory-class classification-factory-class;
 classification-script classification-script;
 congestion-point-profile-script congestion-point-profile-script;
 extension-path extension-path;
}
shared admission-control device name {
 description description;
}
shared admission-control device name interface name {
 description description;
 upstream-provisioned-rate upstream-provisioned-rate;
```

```

downstream-provisioned-rate downstream-provisioned-rate;
upstream-background-bandwidth upstream-background-bandwidth;
downstream-background-bandwidth downstream-background-bandwidth;
action-type (url | python | java-class | java-archive);
action-class-name action-class-name;
action-file-url action-file-url;
action-parameters [action-parameters...];
action-file-name action-file-name;
detect-link-rate;
}
shared congestion-points profile name {
 interface [interface...];
}
slot number acp {
 java-runtime-environment java-runtime-environment;
 java-heap-size java-heap-size;
 java-garbage-collection-options java-garbage-collection-options;
 base-dn base-dn;
 snmp-agent;
 shared shared;
}
slot number acp initial {
 static-dn static-dn;
 dynamic-dn dynamic-dn;
}
slot number acp initial directory-connection {
 url url;
 backup-urls [backup-urls...];
 principal principal;
 credentials credentials;
 protocol (ldaps);
 timeout timeout;
 check-interval check-interval;
 blacklist;
 snmp-agent;
}
slot number acp initial directory-eventing {
 eventing;
 signature-dn signature-dn;
 polling-interval polling-interval;
 event-base-dn event-base-dn;
 dispatcher-pool-size dispatcher-pool-size;
}

```

- Related Topics**
- For detailed information about each configuration statement, see the *SRC PE CLI Command Reference*.
  - Configuring SRC ACP (SRC CLI) on page 236
  - Configuring the SAE for SRC ACP (SRC CLI) on page 241
  - Configuring SRC ACP Properties (SRC CLI) on page 243

## Configuring SRC ACP (SRC CLI)

---

To use SRC ACP in an SRC network, perform the following configuration tasks:

1. (Optional) “Creating Grouped Configurations for SRC ACP (SRC CLI)” on page 236
2. Configuring Local Properties for SRC ACP (SRC CLI) on page 237
3. Configuring the SAE for SRC ACP (SRC CLI) on page 241
4. Configuring SRC ACP Properties (SRC CLI) on page 243
5. (Edge and dual mode only) “Configuring SRC ACP to Manage the Edge Network (SRC CLI)” on page 255
6. (Backbone and dual mode only) “Configuring SRC ACP to Manage the Backbone Network (SRC CLI)” on page 259
7. Starting SRC ACP on page 279

You can automate and scale the configuration of congestion points using congestion point classification. For more information, see “Classifying Congestion Points (SRC CLI)” on page 270.

- Related Topics**
- Configuring SRC ACP (C-Web Interface)
  - Viewing SNMP Information for SRC ACP (SRC CLI) on page 292
  - Configuration Statements for SRC ACP on page 233
  - Overview of SRC ACP on page 223

## Creating Grouped Configurations for SRC ACP (SRC CLI)

---

We recommend that you configure SRC ACP within a group. When you create a configuration group, the software creates a configuration with default values filled in.

Configuration groups allow you to share the SRC ACP configuration with different SRC ACP instances in the SRC network. You can also set up different configurations for different instances.

You can then create a grouped SRC ACP configuration that is shared with some SRC ACP instances. For example, if you create two different SRC ACP groups called `config1` and `config2` within the shared SRC ACP configuration, you could select the SRC ACP configuration that should be associated with a particular SRC ACP instance.

Use the **shared** option of the `slot number acp` statement to select the group for an SRC ACP instance as part of the local configuration. Use the **shared acp group *name*** statements to configure the group.

To select and configure a group:

1. From configuration mode, select a group for an SRC ACP instance. For example, to select a group called `config1` in the path `/:`

```
[edit]
user@host# set slot 0 acp shared /config1
```

For more information, see “Configuring Local Properties for SRC ACP (SRC CLI)” on page 237.

2. Commit the configuration.

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

3. From configuration mode, configure a group. For example, to configure a group called config1, specify the group as part of the SRC ACP configuration.

```
[edit]
user@host# edit shared acp group config1 ?
Possible completions:
<[Enter]> Execute this command
> configuration
> congestion-point-classifier
> group Group of ACP configuration properties
| Pipe through a command
```

For more information, see “Configuring SRC ACP (SRC CLI)” on page 236.

- Related Topics**
- Configuring the SAE for SRC ACP (SRC CLI) on page 241
  - Configuring SRC ACP Properties (SRC CLI) on page 243
  - Creating an Application to Update Information for SRC ACP on page 231
  - Configuration Statements for SRC ACP on page 233
  - Interactions Between SRC ACP and Other Components on page 228

## Configuring Local Properties for SRC ACP (SRC CLI)

Configure initial properties, including Java heap memory, including directory connection and directory eventing properties.

Tasks to configure the local properties for SRC ACP are:

- Configuring Basic Local Properties for SRC ACP on page 237
- Configuring Initial Properties for SRC ACP on page 238
- Configuring Directory Connection Properties for SRC ACP on page 239
- Configuring Initial Directory Eventing Properties for SRC ACP on page 240

### Configuring Basic Local Properties for SRC ACP

Use the following configuration statements to configure basic local properties for SRC ACP:

```
slot number acp {
```

```
java-runtime-environment java-runtime-environment;
java-heap-size java-heap-size;
java-garbage-collection-options java-garbage-collection-options;
base-dn base-dn;
snmp-agent;
shared shared;
}
```

To configure basic local properties:

1. From configuration mode, access the configuration statement that configures the local properties.

```
user@host# edit slot 0 acp
```

2. Specify the basic local properties for ACP.

```
[edit slot 0 acp]
user@host# set ?
```

For more information about configuring local properties for the SRC components, see [Configuring Basic Local Properties](#).

3. Configure the garbage collection functionality of the Java Virtual Machine.

```
[edit slot 0 acp]
user@host# set java-garbage-collection-options java-garbage-collection-options
```

4. Select an SRC ACP group configuration.

```
[edit slot 0 acp]
user@host# set shared shared
```

For more information, see “Creating Grouped Configurations for SRC ACP (SRC CLI)” on page 236.

5. (Optional) Verify your configuration.

```
[edit slot 0 acp]
user@host# show
shared /config;
initial {
 directory-connection {
 url ldap://127.0.0.1:389/
 principal cn=conf,o=Operators,<base>;
 credentials *****;
 }
 directory-eventing {
 eventing;
 polling-interval 30;
 }
}
```

## Configuring Initial Properties for SRC ACP

Use the following configuration statements to configure initial properties for SRC ACP:

```
slot number acp initial {
 static-dn static-dn;
 dynamic-dn dynamic-dn;
}
```

To configure initial local properties:

1. From configuration mode, access the configuration statement that configures the initial properties.

```
user@host# edit slot 0 acp initial
```

2. Specify the properties for SRC ACP.

```
[edit slot 0 acp initial]
user@host# set ?
```

For more information about configuring local properties for the SRC components, see [Configuring Basic Local Properties](#).

3. (Optional) Verify your configuration.

```
[edit slot 0 acp initial]
user@host# show
```

## Configuring Directory Connection Properties for SRC ACP

Use the following configuration statements to configure directory connection properties for SRC ACP:

```
slot number acp initial directory-connection {
 url url;
 backup-urls [backup-urls...];
 principal principal;
 credentials credentials;
 protocol (ldaps);
 timeout timeout;
 check-interval check-interval;
 blacklist;
 snmp-agent;
}
```

To configure directory connection properties:

1. From configuration mode, access the configuration statement that configures the directory connection properties.

```
user@host# edit slot 0 acp initial directory-connection
```

2. Specify the properties for ACP.

```
[edit slot 0 acp initial directory-connection]
user@host# set ?
```

For more information about configuring local properties for the SRC components, see [Configuring Basic Local Properties](#).

3. (Optional) Verify your configuration.

```
[edit slot 0 acp initial directory-connection]
user@host# show
url ldap://127.0.0.1:389/;
principal cn=conf,o=Operators,<base>;
credentials *****;
```

## Configuring Initial Directory Eventing Properties for SRC ACP

Use the following configuration statements to configure directory eventing properties for SRC ACP:

```
slot number acp initial directory-eventing {
 eventing;
 signature-dn signature-dn;
 polling-interval polling-interval;
 event-base-dn event-base-dn;
 dispatcher-pool-size dispatcher-pool-size;
}
```

To configure initial directory eventing properties:

1. From configuration mode, access the configuration statement that configures the local properties.

```
user@host# edit slot 0 acp initial eventing
```

2. Specify the initial directory eventing properties for SRC ACP.

```
[edit slot 0 acp initial directory-eventing]
user@host# set ?
```

For more information about configuring local properties for the SRC components, see [Configuring Basic Local Properties](#).

3. (Optional) Verify your configuration.

```
[edit slot 0 acp initial directory-eventing]
user@host# show
eventing;
polling-interval 30;
```

### Related Topics

- [Configuring SRC ACP \(SRC CLI\) on page 236](#)
- [Creating Grouped Configurations for SRC ACP \(SRC CLI\) on page 236](#)
- [Configuring Local Properties for SRC ACP \(C-Web Interface\)](#)
- [Configuring SRC ACP Properties \(SRC CLI\) on page 243](#)



## Configuring the SAE for SRC ACP (SRC CLI)

You must configure the SAE to recognize SRC ACP by adding information about SRC ACP to the SAE properties. The tasks for configuring the SAE for SRC ACP are:

- Configuring SRC ACP as an External Plug-In on page 241
- Configuring Event Publishers on page 241
- Configuring the SAE to Monitor Interfaces for Congestion Points on page 241

### Configuring SRC ACP as an External Plug-In

To configure an external plug-in for the SAE:

1. From configuration mode, access the configuration statement that configures the external plug-ins.

```
user@host# edit shared sae configuration plug-ins name name external
```

2. Specify the the plug-in attributes.

```
[edit shared sae configuration plug-ins name name external]
user@host# set attributes ?
```

For edge and dual modes—upstream-bandwidth, downstream-bandwidth, service-name, router-name, login-name, user-dn, port-id, session-id, user-ip-address, nas-ip, user-session-id, event-time

For backbone mode—upstream-bandwidth, downstream-bandwidth, service-name, router-name, session-id, nas-ip, event-time

For more information about configuring plug-in attributes, see *Configuring the SAE for External Plug-Ins (SRC CLI)*.

### Configuring Event Publishers

You must configure the SAE to publish the following types of events to SRC ACP:

- (Edge and dual mode only) Global subscriber tracking
- Global service authorization
- Global service tracking

For information about configuring event publishers, see *Special Types of Event Publishers*. Identify the instance of SRC ACP by the name of the host on which you configured it.

### Configuring the SAE to Monitor Interfaces for Congestion Points



**NOTE:** Configure this feature only if SRC ACP is in backbone or dual mode.

The SAE uses a hosted internal plug-in to monitor the state of interfaces on a VR for backbone congestion points. If a subscriber tries to activate a service on an interface that

is unavailable, the SAE denies the request. The plug-in also monitors the directory for new backbone congestion points.

When this plug-in initializes, it reads all the backbone services from the directory and generates a list of the DNs (network interfaces) of the backbone congestion points. The SAE sends interface tracking events, which contain the names of the interfaces, VRs, and routers to this plug-in. For this feature to work correctly, the interface, VR, and router must be configured (see “Configuring Network Interfaces in the Directory for the Backbone Network” on page 259).

To configure the ACP interface listener as an internal plug-in for the SAE:

1. From configuration mode, access the configuration statement that configures the ACP interface listener.

```
user@host# edit shared sae configuration plug-ins name name acp-interface-listener
```

2. Specify the IP address or name of the host that supports the directory that contains backbone service definitions and network interfaces.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# set ldap-server ldap-server
```

3. Specify the DN of the directory entry that defines the username with which the plug-in accesses the directory.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# set bind-dn bind-dn
```

4. Specify the password with which the plug-in accesses the directory.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# set bind-password bind-password
```

5. Specify whether the connection to the directory uses secure LDAP. If you do not configure a security protocol, plain socket is used.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# set ldaps
```

6. Specify the DN at which SRC ACP stores backbone congestion points.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# set congestion-points-base-dn congestion-points-base-dn
```

7. Specify the DN at which SRC ACP stores edge congestion points.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# set admission-control-base-dn admission-control-base-dn
```

8. (Optional) Specify the maximum time that the plug-in waits for the router to respond.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# set timeout timeout
```

9. Specify the object reference for the ACP plug-in, as defined by the object reference for SRC ACP (see information about the **acp-ior** option in “Configuring SRC ACP Properties (SRC CLI)” on page 243).

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# set acp-remote-corba-ior acp-remote-corba-ior
```

10. (Optional) Verify your configuration.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# show
```

- Related Topics**
- Configuring the SAE for SRC ACP (C-Web Interface)
  - Configuring SRC ACP (SRC CLI) on page 236
  - Configuring SRC ACP to Manage the Edge Network (SRC CLI) on page 255
  - Configuring SRC ACP to Manage the Backbone Network (SRC CLI) on page 259

## Configuring SRC ACP Properties (SRC CLI)

To configure SRC ACP properties, perform these tasks:

1. Configuring Logging Destinations for SRC ACP on page 243
2. Configuring SRC ACP Operation on page 244
3. Configuring CORBA Interfaces on page 248
4. Configuring SRC ACP Redundancy on page 249
5. Configuring Connections to the Subscribers' Directory on page 250
6. Configuring Connections to the Services' Directory on page 252
7. Configuring SRC ACP Scripts and Classification on page 254

## Configuring Logging Destinations for SRC ACP

Use the following configuration statements to configure logging destinations for SRC ACP:

```
shared acp configuration logger name ...
shared acp configuration logger name file {
 filter filter;
 filename filename;
 rollover-filename rollover-filename;
 maximum-file-size maximum-file-size;
}
shared acp configuration logger name syslog {
 filter filter;
 host host;
 facility facility;
 format format;
}
```

**Configuring Logging Destinations to Store Messages in a File**

To configure logging destinations to store log messages in a file:

1. From configuration mode, access the configuration statement that configures the name and type of logging destination. In this sample procedure, the logging destination called file-1 is configured in the config group.

```
user@host# edit shared acp group config configuration logger file-1 file
```

2. Specify the properties for the logging destination.

```
[edit shared acp group config configuration logger file-1 file]
user@host# set ?
```

For more information about configuring properties for the logging destination, see [Configuring a Component to Store Log Messages in a File \(SRC CLI\)](#).

3. (Optional) Verify your configuration.

```
[edit shared acp group config configuration logger file-1 file]
user@host# show
filename var/log/acp_debug.log;
rollover-filename var/log/acp_debug.alt;
```

**Configuring Logging Destinations to Send Messages to System Logging Facility**

To configure logging destinations to send log messages to the system logging facility:

1. From configuration mode, access the configuration statement that configures the name and type of logging destination. In this sample procedure, the logging destination called syslog-1 is configured in the config group.

```
user@host# edit shared acp group config configuration logger syslog-1 syslog
```

2. Specify the properties for the logging destination.

```
[edit shared acp group config configuration logger syslog-1 syslog]
user@host# set ?
```

For more information about configuring properties for the logging destination, see [Configuring System Logging \(SRC CLI\)](#).

3. (Optional) Verify your configuration.

```
[edit shared acp group config configuration logger syslog-1 syslog]
user@host# show
filter /error-;
host loghost;
```

## Configuring SRC ACP Operation

Use the following configuration statements to configure how SRC ACP operates:

```
shared acp configuration acp-options {
 backup-directory backup-directory;
 mode (edge | backbone | dual);
 event-cache-size event-cache-size;
 overload-method overload-method;
```

```

reservation-timeout reservation-timeout;
congestion-point-auto-completion;
tuning-factor tuning-factor;
subscriber-bandwidth-exceed-message subscriber-bandwidth-exceed-message;
network-bandwidth-exceed-message network-bandwidth-exceed-message;
backup-database-maximum-size backup-database-maximum-size;
remote-update-database-index-keys remote-update-database-index-keys;
interface-tracking-filter interface-tracking-filter;
state-sync-bulk-size state-sync-bulk-size;
}

```

To configure SRC ACP operation:

1. From configuration mode, access the configuration statement that configures SRC ACP operation. In this sample procedure, the SRC ACP operating properties are configured in the `config` group.

```
user@host# edit shared acp group config configuration acp-options
```

2. Specify the folder that stores backup information about subscribers, services, and congestion points.

```
[edit shared acp group config configuration acp-options]
user@host# set backup-directory
```

3. Specify the regions of the network that SRC ACP manages.

```
[edit shared acp group config configuration acp-options]
user@host# set mode (edge | backbone | dual)
```

4. Specify the number of plug-in events from the SAE that SRC ACP can store in its cache.

```
[edit shared acp group config configuration acp-options]
user@host# set event-cache-size event-cache-size
```

5. Specify how SRC ACP deals with situations in which the components exceed the allocated bandwidth because the service was activated after the authorization was granted.

```
[edit shared acp group config configuration acp-options]
user@host# set overload-method overload-method
```

If you specify `-1`, SRC ACP ignores overload. An integer greater than or equal to 0 specifies the bandwidth (in bits per second) by which the maximum may be exceeded.

6. Specify the time to wait before a bandwidth reservation expires. The reserved bandwidth is reclaimed by SRC ACP when the reservation expires.

```
[edit shared acp group config configuration acp-options]
user@host# set reservation-timeout reservation-timeout
```

7. Specify whether SRC ACP uses the information acquired from the router to determine the congestion points.

```
[edit shared acp group config configuration acp-options]
user@host# set congestion-point-auto-completion
```

8. Specify the factors that compensate for actual use of bandwidth, as opposed to allocated bandwidth.

```
[edit shared acp group config configuration acp-options]
user@host# set tuning-factor tuning-factor
```

9. Specify the error message that SRC ACP sends when the subscriber exceeds the allocated bandwidth.

```
[edit shared acp group config configuration acp-options]
user@host# set subscriber-bandwidth-exceed-message
subscriber-bandwidth-exceed-message
```

10. Specify the error message that SRC ACP sends when traffic flow exceeds the allocated bandwidth on an interface between the subscriber and the router.

```
[edit shared acp group config configuration acp-options]
user@host# set network-bandwidth-exceed-message
network-bandwidth-exceed-message
```

11. Specify the value by which the sum of the sizes of the files that contain SRC ACP data can increment before SRC ACP reorganizes the files.

```
[edit shared acp group config configuration acp-options]
user@host# set backup-database-maximum-size backup-database-maximum-size
```

Choose a value that is significantly lower than the capacity of the machine's hard disk.

12. Specify the values to look for in the configuration data. Specifying index keys can improve performance by filtering the data.

```
[edit shared acp group config configuration acp-options]
user@host# set remote-update-database-index-keys
remote-update-database-index-keys
```

The value is a list of attributes, separated by commas. An attribute is one of the following text strings:

- accountingId—Value of directory attribute accountingUserId.
- dhcpPacket—Content of the DHCP discover request.
- hostname— Name of the host on which the SAE is installed.
- ifIndex—SNMP index of the interface. This attribute is not supported on JUNOS routing platforms.
- ifRadiusClass—RADIUS class attribute on the JUNOS interface. This attribute is not supported on JUNOS routing platforms.

- `ifSessionId`—Identifier for RADIUS accounting on the JUNOS interface. This attribute is not supported on JUNOS routing platforms.
  - `interfaceAlias`—Alias of the interface; that is, the IP description in the interface configuration.
  - `interfaceDescr`—SNMP description of the interface.
  - `interfaceName`—Name of the interface.
  - `loginName`—Subscriber's login name.
  - `nasInetAddress`—IP address of the router; using a byte array instead of an integer.
  - `nasPort`—NAS port used by the router to identify the interface to RADIUS.
  - `portId`—Identifier of VLAN or virtual circuit. For a virtual circuit, use the format `<VPI>/<VCI>`. This attribute is not supported on JUNOS routing platforms.
    - `<VPI>`—Virtual path identifier
    - `<VCI>`—Virtual connection identifier
  - `primaryUserName`—PPP login name or the public DHCP username. This attribute is not supported on JUNOS routing platforms.
  - `routerName`—Name of the virtual router in the format `<virtualRouter>@<router>`.
    - `<virtualRouter>`—Virtual router name
    - `<router>`—Router name
  - `routerType`—Type of router driver.
  - `userInetAddress`—IP address of the subscriber that uses a byte array instead of an integer.
  - `userMacAddress`—MAC address of the DHCP subscriber. This attribute is not supported on JUNOS routing platforms.
  - `userRadiusClass`—RADIUS class attribute of the subscriber session for a service. This attribute can occur multiple times and can be returned by an authorization plug-in.
  - `userType`—Type of subscriber.
13. Specify the interface tracking event to be ignored by SRC ACP.

```
[edit shared acp group config configuration acp-options]
user@host# set interface-tracking-filter interface-tracking-filter
```

The value is filter strings in the format of a list of `<attribute>=<value>` pairs. The filter strings can be contained within query operations.

- <attribute>—Name of an attribute for an interface tracking event. See value for the **remote-update-database-index-keys** option described “Configuring SRC ACP Properties (SRC CLI)” on page 243.
  - <value>—Filtering string of the following types:
    - \*—Any value
    - Explicit string—Any value matching the specified string (not case-sensitive)
    - String containing an asterisk—Any value containing the specified string (not case-sensitive)
  - To perform query operations on filter strings, you can use the following values in your filter strings:
    - ()—Match no objects.
    - (\*)—Match all objects.
    - (&<filter><filter>...)—Performs logical AND operation on filter strings; true if all filter strings match.
    - (|<filter><filter>...)—Performs logical OR operation on filter strings; true if at least one filter string matches.
    - (!<filter>)—Performs logical NOT operation on filter string; true if the filter string does not match.
14. (Optional) Specify the number of events the SAE sends to SRC ACP in a single method call during state synchronization.
- ```
[edit shared acp group config configuration acp-options]
user@host# set state-sync-bulk-size state-sync-bulk-size
```
15. (Optional) Verify your configuration.
- ```
[edit shared acp group config configuration acp-options]
user@host# show
```

## Configuring CORBA Interfaces

Use the following configuration statements to configure CORBA interfaces for SRC ACP:

```
shared acp configuration corba {
 acp-ior acp-ior;
 remote-update-ior remote-update-ior;
}
```

To configure CORBA interfaces:

1. From configuration mode, access the configuration statement that configures CORBA interfaces for SRC ACP. In this sample procedure, the CORBA interfaces are configured in the config group.
- ```
user@host# edit shared acp group config configuration corba
```


2. Export the object reference for SRC ACP through either a local file or a Common Object Services (COS) naming service.

```
[edit shared acp group config configuration corba]
user@host# set acp-ior acp-ior
```

3. Specify the object reference for the ACP external interface.

```
[edit shared acp group config configuration corba]
user@host# set remote-update-ior remote-update-ior
```

4. (Optional) Verify your configuration.

```
[edit shared acp group config configuration corba]
user@host# show
  acp-ior file:///var/acp/acp.ior;
  remote-update-ior file:///var/acp/sra.ior;
```

Configuring SRC ACP Redundancy

Use the following configuration statements to configure SRC ACP redundancy and state synchronization with the SAE:

```
shared acp configuration redundancy {
  enable-redundancy;
  local-ior local-ior;
  remote-ior remote-ior;
  ignore-user-tracking-out-of-sync;
  community-heartbeat community-heartbeat;
  community-acquire-timeout community-acquire-timeout;
  community-blackout-timeout community-blackout-timeout;
  redundant-naming-service redundant-naming-service;
}
```

To configure SRC ACP redundancy and state synchronization with the SAE:

1. From configuration mode, access the configuration statement that configures SRC ACP redundancy. In this sample procedure, the properties are configured in the config group.

```
user@host# edit shared acp group config configuration redundancy
```

2. (Optional) Enable SRC ACP redundancy.

```
[edit shared acp group config configuration redundancy]
user@host# set enable-redundancy
```

3. Export the object reference for this SRC ACP (local interface) through a Common Object Services (COS) naming service in a redundant SRC ACP configuration.

```
[edit shared acp group config configuration redundancy]
user@host# set local-ior local-ior
```

4. Resolves the object reference for the other SRC ACP (remote interface) through a Common Object Services (COS) naming service in a redundant SRC ACP

configuration. For redundancy, the remote IOR value of one SRC ACP must match the local IOR value of the other SRC ACP.

```
[edit shared acp group config configuration redundancy]
user@host# set remote-ior remote-ior
```

5. (Optional) Specify whether user tracking events should be ignored when they raise an OutOfSync exception to the SAE when state synchronization is enabled. SRC ACP raises an OutOfSync exception when SRC ACP handles service tracking or authentication events without receiving a user start event first.

```
[edit shared acp group config configuration redundancy]
user@host# set ignore-user-tracking-out-of-sync
```

6. (Optional) Specify the time interval for community members to check each other's availability when both redundancy and state synchronization are enabled.

```
[edit shared acp group config configuration redundancy]
user@host# set community-heartbeat community-heartbeat
```

7. (Optional) Specify the time to wait before trying to reacquire the distributed lock when both redundancy and state synchronization are enabled.

```
[edit shared acp group config configuration redundancy]
user@host# set community-acquire-timeout community-acquire-timeout
```

8. (Optional) Specify the time to wait before regaining control when both redundancy and state synchronization are enabled.

```
[edit shared acp group config configuration redundancy]
user@host# set community-blackout-timeout community-blackout-timeout
```

9. Export the object reference for the backup naming service through a local file or COS naming service in a redundant SRC ACP configuration. The primary SRC ACP registers the IOR and redundancy IOR to both naming services, while the secondary SRC ACP registers the redundancy IOR to both naming services.

```
[edit shared acp group config configuration redundancy]
user@host# set redundant-naming-service redundant-naming-service
```

10. (Optional) Verify your configuration.

```
[edit shared acp group config configuration redundancy]
user@host# show
```

Configuring Connections to the Subscribers' Directory

Use the following configuration statements to configure how SRC ACP connects to the directory that contains subscriber information:

```
shared acp configuration ldap subscriber-data {
  congestion-points-eventing;
  server-address server-address;
  server-port server-port;
```

```

dn dn;
principal principal;
password password;
event-dn event-dn;
directory-eventing;
polling-interval polling-interval;
}

```

To configure connections to the directory that stores subscriber information:

1. From configuration mode, access the configuration statement that configures SRC ACP connections to the subscribers' directory. In this sample procedure, the connections are configured in the config group.

```

user@host# edit shared acp group config configuration ldap subscriber-data

```

2. (Optional) Enable directory eventing for congestion points.

```

[edit shared acp group config configuration ldap subscriber-data]
user@host# set congestion-points-eventing

```

3. Specify the list of primary and redundant servers that manage data for subscribers.

```

[edit shared acp group config configuration ldap subscriber-data]
user@host# set server-address server-address

```

4. Specify the TCP port for the directory.

```

[edit shared acp group config configuration ldap subscriber-data]
user@host# set server-port server-port

```

5. Specify the DN of the root of the directory.

```

[edit shared acp group config configuration ldap subscriber-data]
user@host# set dn dn

```

6. Specify the DN used to authorize connections to the directory.

```

[edit shared acp group config configuration ldap subscriber-data]
user@host# set principal principal

```

7. Specify the password used to authorize connections to the directory.

```

[edit shared acp group config configuration ldap subscriber-data]
user@host# set password password

```

8. Specify the DN of the directory that contains event information.

```

[edit shared acp group config configuration ldap subscriber-data]
user@host# set event-dn event-dn

```

9. (Optional) Enable directory eventing.

```

[edit shared acp group config configuration ldap subscriber-data]
user@host# set directory-eventing

```

10. Specify the time interval at which the SRC component polls the directory.

```
[edit shared acp group config configuration ldap subscriber-data]
user@host# set polling-interval polling-interval
```

11. (Optional) Verify your configuration.

```
[edit shared acp group config configuration ldap subscriber-data]
user@host# show
```

Configuring Connections to the Services' Directory

Use the following configuration statements to configure how SRC ACP connects to the directory that contains information about services:

```
shared acp configuration ldap service-data {
  edge-congestion-point-dn edge-congestion-point-dn;
  backbone-congestion-point-dn backbone-congestion-point-dn;
  reload-congestion-points;
  congestion-points-eventing;
  server-address server-address;
  server-port server-port;
  dn dn;
  principal principal;
  password password;
  event-dn event-dn;
  directory-eventing;
  polling-interval polling-interval;
}
```

To configure connections to the directory that stores service information:

1. From configuration mode, access the configuration statement that configures SRC ACP connections to the services' directory. In this sample procedure, the connections are configured in the config group.

```
user@host# edit shared acp group config configuration ldap service-data
```

2. Specify the DN of the directory that contains information about network interfaces for edge congestion points.

```
[edit shared acp group config configuration ldap service-data]
user@host# set edge-congestion-point-dn edge-congestion-point-dn
```

3. Specify the DN of the directory that contains information about network interfaces for backbone congestion point objects.

```
[edit shared acp group config configuration ldap service-data]
user@host# set backbone-congestion-point-dn backbone-congestion-point-dn
```

4. (Optional) Specify whether SRC ACP detects changes in the backbone congestion point for a service while SRC ACP is operative.

```
[edit shared acp group config configuration ldap service-data]
user@host# set reload-congestion-points
```

Set this value only when you want to modify a congestion point.

5. (Optional) Enable directory eventing for congestion points.
[edit shared acp group config configuration ldap service-data]
user@host# **set congestion-points-eventing**
6. Specify the list of primary and redundant servers that manage data for subscribers.
[edit shared acp group config configuration ldap service-data]
user@host# **set server-address** *server-address*
7. Specify the TCP port for the directory.
[edit shared acp group config configuration ldap service-data]
user@host# **set server-port** *server-port*
8. Specify the DN of the root of the directory.
[edit shared acp group config configuration ldap service-data]
user@host# **set dn** *dn*
9. Specify the DN used to authorize connections to the directory.
[edit shared acp group config configuration ldap service-data]
user@host# **set principal** *principal*
10. Specify the password used to authorize connections to the directory.
[edit shared acp group config configuration ldap service-data]
user@host# **set password** *password*
11. Specify the DN of the directory that contains event information.
[edit shared acp group config configuration ldap service-data]
user@host# **set event-dn** *event-dn*
12. (Optional) Enable directory eventing.
[edit shared acp group config configuration ldap service-data]
user@host# **set directory-eventing**
13. Specify the time interval at which the SRC component polls the directory.
[edit shared acp group config configuration ldap service-data]
user@host# **set polling-interval** *polling-interval*
14. (Optional) Verify your configuration.
[edit shared acp group config configuration ldap service-data]
user@host# **show**

Configuring SRC ACP Scripts and Classification

Use the following configuration statements to configure SRC ACP scripts and classification:

```
shared acp configuration scripts-and-classification {  
  script-factory-class script-factory-class;  
  classification-factory-class classification-factory-class;  
  classification-script classification-script;  
  congestion-point-profile-script congestion-point-profile-script;  
  extension-path extension-path;  
}
```

To configure scripts and classification:

1. From configuration mode, access the configuration statement that configures SRC ACP scripts and classification. In this sample procedure, the properties are configured in the config group.

```
user@host# edit shared acp group config configuration scripts-and-classification
```

2. Specify the script factory class name.

```
[edit shared acp group config configuration scripts-and-classification]  
user@host# set script-factory-class script-factory-class
```

3. Specify the congestion point classifier factory class name.

```
[edit shared acp group config configuration scripts-and-classification]  
user@host# set classification-factory-class classification-factory-class
```

4. Specify the class name for congestion point classification.

```
[edit shared acp group config configuration scripts-and-classification]  
user@host# set classification-script classification-script
```

5. Specify the class name for generating the congestion point DN by using the congestion point profile.

```
[edit shared acp group config configuration scripts-and-classification]  
user@host# set congestion-point-profile-script congestion-point-profile-script
```

6. Specify the extension class path for classes not located in the `/opt/UMC/acp/lib` directory.

```
[edit shared acp group config configuration scripts-and-classification]  
user@host# set extension-path extension-path
```

7. (Optional) Verify your configuration.

```
[edit shared acp group config configuration scripts-and-classification]  
user@host# show
```

Related Topics • [Configuring SRC ACP Properties \(C-Web Interface\)](#)

- Configuring Local Properties for SRC ACP (SRC CLI) on page 237
- Configuring SRC ACP (SRC CLI) on page 236
- Configuring ACP to Store Log Messages in a File (C-Web Interface)

Configuring SRC ACP to Manage the Edge Network (SRC CLI)

The tasks to configure SRC ACP to manage the edge network are:

- Configuring Network Interfaces in the Directory for the Edge Network on page 255
- Configuring Bandwidths for Subscribers on page 256
- Assigning Network Interfaces to Subscribers on page 257
- Configuring Bandwidths for Services in the Edge Network on page 258

Configuring Network Interfaces in the Directory for the Edge Network

You must add network interfaces to the directory. For the edge network, you do so by specifying the network interfaces of the routers and the switches in the access network between subscribers and the SRC network.

Use the following configuration statements to configure a network interface:

```
shared admission-control device name {
    description description;
}
shared admission-control device name interface name {
    description description;
    upstream-provisioned-rate upstream-provisioned-rate;
    downstream-provisioned-rate downstream-provisioned-rate;
    upstream-background-bandwidth upstream-background-bandwidth;
    downstream-background-bandwidth downstream-background-bandwidth;
    detect-link-rate;
}
```

To configure the network interfaces of the routers and the switches in the access network:

1. From configuration mode, access the configuration statement that configures network interfaces.

```
user@host# edit shared admission-control device name
```

Enter the name of the network device.

2. (Optional) Specify a description for the network device.

```
[edit shared admission-control device name]
user@host# set description description
```

3. Specify the network interface.

```
user@host# edit shared admission-control device name interface name
```

Enter the name of the virtual router.

4. (Optional) Specify the provisioned bandwidth for the network interface.

```
[edit shared admission-control device name interface name]  
user@host# set upstream-provisioned-rate upstream-provisioned-rate  
user@host# set downstream-provisioned-rate downstream-provisioned-rate
```

5. (Optional) Specify the background bandwidth for the network interface.

```
[edit shared admission-control device name interface name]  
user@host# set upstream-background-bandwidth upstream-background-bandwidth  
user@host# set downstream-background-bandwidth  
          downstream-background-bandwidth
```

For information about background bandwidths, see “Allocating Bandwidth to Applications Not Controlled by SRC ACP” on page 227.

6. (Optional) Specify whether SRC ACP detects the link rate for the network interface.

```
[edit shared admission-control device name interface name]  
user@host# set detect-link-rate
```

If you set this option, specify portId as an index key when configuring SRC ACP operations so that updated sync rates are provided from interface tracking events. If the sync rate is not available, then the provisioned bandwidth configured in the subscriber profile is used.

7. (Optional) Verify your configuration.

```
[edit shared admission-control device name interface name]  
user@host# show
```

Configuring Bandwidths for Subscribers

You must configure bandwidths for subscribers that SRC ACP manages in the edge region of the network.

If the access network between the subscriber and the router uses ATM, and all the traffic coming from one DSLAM travels on a single virtual path, you do not need to provision bandwidths for each subscriber. In this case, SRC ACP can derive the congestion points from the router (see “Deriving Congestion Points Automatically” on page 225).

However, if the access network uses a protocol other than ATM, you must provide the following information for each subscriber.

- Provisioned downstream bandwidth
- Provisioned upstream bandwidth
- Actual downstream bandwidth for the current subscriber session
- Actual upstream bandwidth for the current subscriber session
- List of DNS of interfaces associated with congestion points

To configure bandwidths for subscribers:

1. From configuration mode, access the configuration statement that configures residential subscribers.

```
user@host# edit subscribers retailer name subscriber-folder folder-name subscriber
name admission-control
```

For more information about configuring residential subscribers, see Adding Residential Subscribers (SRC CLI).

2. (Optional) Specify the provisioned downstream bandwidth. This rate is used if the subscriber bandwidth settings are not provided by remote update (through the API for ACP) or by the **downstream-sync-rate** value.

```
[edit subscribers retailer name subscriber-folder folder-name subscriber name
admission-control]
user@host# set downstream-provisioned-rate downstream-provisioned-rate
```

3. (Optional) Specify the provisioned upstream bandwidth. This rate is used if the subscriber bandwidth settings are not provided by remote update (through the API for ACP) or by the **upstream-sync-rate** value.

```
[edit subscribers retailer name subscriber-folder folder-name subscriber name
admission-control]
user@host# set upstream-provisioned-rate upstream-provisioned-rate
```

4. (Optional) Specify the actual downstream bandwidth for the current subscriber session. If you do not set this value and it is not provided by remote update (through the API for ACP), then the **downstream-provisioned-rate** value is used.

```
[edit subscribers retailer name subscriber-folder folder-name subscriber name
admission-control]
user@host# set downstream-sync-rate downstream-sync-rate
```

5. (Optional) Specify the actual upstream bandwidth for the current subscriber session. If you do not set this value and it is not provided by remote update (through the API for ACP), then the **upstream-provisioned-rate** value is used.

```
[edit subscribers retailer name subscriber-folder folder-name subscriber name
admission-control]
user@host# set upstream-sync-rate upstream-sync-rate
```

Assigning Network Interfaces to Subscribers

You must assign to the subscriber object interfaces (including the router interfaces) for all congestion points between the subscriber and the router.



NOTE: You must define the interface in the directory before you can assign it to a residential subscriber (see “Configuring Network Interfaces in the Directory for the Edge Network” on page 255).

To assign an interface:

1. From configuration mode, access the configuration statement that configures residential subscribers.

```
user@host# edit subscribers retailer name subscriber-folder folder-name subscriber
name admission-control
```

For more information about configuring residential subscribers, see Adding Residential Subscribers (SRC CLI).

2. (Optional) Specify the DNs of interfaces associated with congestion points for this subscriber.

```
[edit subscribers retailer name subscriber-folder folder-name subscriber name
admission-control]
user@host# set congestion-points [congestion-points...]
```

Configuring Bandwidths for Services in the Edge Network

Upstream and downstream bandwidths must be specified for services that SRC ACP manages. You can obtain bandwidths for services in two ways:

- Provide static values through the directory.
- Allow the values to be provided through the SAE core API.

For example, a business partner may need to specify the required values for a particular piece of content through the SAE core API.

To configure values for services:

1. From configuration mode, access the configuration statement that configures services.

```
user@host# edit services global service name admission-control
```

For more information about configuring services, see Overview of Services for the SRC Software.

2. (Optional) Specify the required downstream and upstream bandwidths.

```
[edit services global service name admission-control]
user@host# set required-downstream-bandwidth required-downstream-bandwidth
user@host# set required-upstream-bandwidth required-upstream-bandwidth
```

Related Topics

- Configuring SRC ACP to Manage the Edge Network (C-Web Interface)
- Configuring SRC ACP to Manage the Backbone Network (SRC CLI) on page 259
- Viewing Information About Subscriber Sessions in the Edge Network (SRC CLI) on page 283
- Overview of SRC ACP on page 223

Configuring SRC ACP to Manage the Backbone Network (SRC CLI)

The tasks to configure SRC ACP to manage the backbone network are:

- Configuring Network Interfaces in the Directory for the Backbone Network on page 259
- Extending SRC ACP Congestion Points for the Backbone Network on page 259
- Configuring Action Congestion Points on page 260
- Configuring Bandwidths for Services in the Backbone Network on page 261
- Configuring Congestion Points for Services in the Backbone Network on page 261
- Using Functions for Backbone Congestion Point Classification Scripts on page 266
- Configuring Congestion Point Profiles in the Directory on page 267
- Assigning Interfaces to Congestion Point Profiles on page 267

Configuring Network Interfaces in the Directory for the Backbone Network

You configure network interfaces in the directory in the same way for edge and backbone congestion points.

- For backbone congestion points, add only VRs and their interfaces. For information about this procedure, see “Configuring Network Interfaces in the Directory for the Edge Network” on page 255.

Extending SRC ACP Congestion Points for the Backbone Network

You can extend SRC ACP congestion points to initialize and execute applications defined in a backbone congestion point.

SRC ACP provides a service provider interface (SPI) to:

- Create custom congestion point applications that authorize service activation and track service start and stop events.
- Obtain congestion point information from remote update.
- Retrieve congestion point status.
- Track congestion point state.

The SPI for ACP provides a Java interface that a congestion point application implements. For information about the SPI for ACP, see the SDK documentation in the SDK+AppSupport+Demos+Samples.tar.gz file on the Juniper Networks Web site at: <https://www.juniper.net/support/csc/swdist-erx/src.html> You can locate the files in the SDK/doc/acp directory.

The implementation of the SPI for ACP can be a customized application that performs certain tasks, such as creating or removing congestion points on the router. SRC ACP acts as an interface tracking plug-in, and interface tracking events are treated as remote updates for congestion points when they are created, modified, or removed.

SRC ACP supports applications written in Java or Jython. For scripts written in Java, you must compile and package the implemented SPI for ACP to make it available for use by SRC ACP. A Java implementation can include more than one Java archive (JAR) file.

To use congestion point applications with SRC ACP, configure an action congestion point that references the script.

Configuring Action Congestion Points

You can define an application in a backbone congestion point so that SRC ACP can execute it in a predefined manner. Backbone congestion points that are configured to run an application are called action congestion points. If you want to use an action congestion point to execute an application that requires real-time congestion point status, you must enable SRC ACP state synchronization with the SAE).

Before you configure an action congestion point, make sure that you know the location of the application file.

Use the following configuration statements to configure action congestion points:

```
shared admission-control device name interface name {  
  action-type (url | python | java-class | java-archive);  
  action-class-name action-class-name;  
  action-file-url action-file-url;  
  action-parameters [action-parameters...];  
  action-file-name action-file-name;  
}
```

To configure an action congestion point:

1. From configuration mode, access the configuration statement that configures network interfaces.

```
user@host# edit shared admission-control device name interface name
```

Enter the name of the network device and the name of the virtual router.

2. (Optional) Specify the file type of the application.

```
[edit shared admission-control device name interface name]  
user@host# set action-type (url | python | java-class | java-archive);
```

3. (Optional) Specify the name of the class implementing the SPI.

```
[edit shared admission-control device name interface name]  
user@host# set action-class-name action-class-name
```

4. (Optional) Specify the URL or the content of the file. For action congestion point implementations written in Java of the url action type, configure the URL that specifies the location of the Java archives (.jar files) containing the action congestion point implementation. For other action types, you must load the action congestion point implementation with the action file name option.

```
[edit shared admission-control device name interface name]  
user@host# set action-file-url action-file-url
```

5. (Optional) Specify the parameter as an attribute=value pair.

```
[edit shared admission-control device name interface name]
user@host# set action-parameters [action-parameters...]
```

6. (Optional) Load the local file that contains the action congestion point implementation. This file is the uncompiled Python source code or the compiled result of the Java file (binary *.class* or *.jar* file).

```
[edit shared admission-control device name interface name]
user@host# set action-file-name action-file-name
```

7. (Optional) Verify your configuration.

```
[edit shared admission-control device name interface name]
user@host# show
```

Configuring Bandwidths for Services in the Backbone Network

To configure bandwidths for services in the same way for edge and backbone congestion points:

- See “Configuring SRC ACP to Manage the Edge Network (SRC CLI)” on page 255.

Configuring Congestion Points for Services in the Backbone Network

You must assign a congestion point to each service that SRC ACP manages. When SRC ACP receives a service authorization event, congestion points for a service session can be determined by:

- Congestion point classification
- Congestion point profiles

To configure congestion points with congestion point classification:

1. From configuration mode, access the configuration statement that configures services.

```
user@host# edit services global service name admission-control
congestion-point-classification
```

For more information about services, see Overview of Services for the SRC Software.

2. Specify the backbone congestion point expression.

```
[edit services global service name admission-control congestion-point-classification]
user@host# set expression [expression...]
```

The syntax for a backbone congestion point expression is defined in the format <NetworkDevice>/<NetworkInterface>/<InstanceID> which maps to a congestion point.

- <NetworkDevice>—Network device listed in the directory.

- <NetworkInterface>—Network interface listed in the directory.
- <InstanceID>—Name of an instance of a congestion point that is automatically created.

For information about congestion point expressions, see “Congestion Point Expressions” on page 276. For information about the attributes that can be embedded in the expression, see “Plug-In Attributes for Use with Backbone Congestion Point Expressions” on page 263.

3. (Optional) Specify the backbone congestion point script.

```
[edit services global service name admission-control congestion-point-classification]
user@host# set script script
```

For information about congestion point functions, see “Using Functions for Backbone Congestion Point Classification Scripts” on page 266.

To configure congestion points with congestion point profiles:

1. From configuration mode, access the configuration statement that configures services.

```
user@host# edit services global service name admission-control
```

For more information about services, see Overview of Services for the SRC Software.

2. (Optional) Specify the backbone congestion points. This value is ignored if you configure congestion points with congestion point classification.

```
[edit services global service name admission-control]
user@host# set congestion-points [congestion-points...]
```

The backbone congestion point is defined in the format <-vrName->/<-serviceName->, which locates a congestion point profile that contains a list of congestion points.

- To allow the software to automatically define the congestion point, use the entry <-vrName->/<-serviceName->. When SRC ACP starts operating, it will substitute the VR name and the service name from the request for service activation.
- To restrict the congestion point to a specific VR or service, enter the actual VR name or service name.

Plug-In Attributes for Use with Backbone Congestion Point Expressions

These plug-in attributes must be available for service authorization and service tracking events.

accountingId

- Value of accountingUserId attribute.

ifRadiusClass

- RADIUS class attribute on the JUNOS interface.
- Value—String array
- Example—ifRadiusClass=" acpe"

ifSessionId

- Identifier for RADIUS accounting on the JUNOS interface.

interfaceAlias

- Description of the interface.
- Value—Interface description that is configured on the JUNOS router with the **interface ip description** command
- Example—interfaceAlias=" dhcp-subscriber12"

interfaceDescr

- Alternate name for the interface that is used by SNMP. This name is a system-generated name.
- Value
 - On a JUNOS router, the format of the description is
ip<slot>/<port>.<subinterface>
 - On the JUNOS routing platform, interfaceDescr is the same as interfaceName.
- Example—interfaceDescr=" IP3/1"

interfaceName

- Name of the interface.
- Value
 - Name of the interface in your router CLI syntax
 - FORWARDING_INTERFACE for routing instance (used by traffic mirroring)
- Example—For JUNOSe routers: interfaceName=" fastEthernet6/0"
For JUNOS routing platforms: interfaceName="fe-0/1/0.0"
For forwarding interface: interfaceName="FORWARDING_INTERFACE"

localQosProfiles.<layer name>

- Local QoS profile in the specified layer. Local QoS profiles refer to profiles that are attached using the JunosE router CLI or the Service Manager and not through a SAE.
- Value—String
 - The <layer name> is one of the following values: ip, ipv6, lac, svlan, vlan, ethernet, atmVp, atmVc, atm, bridge, frVc, ipTunnel, l2tpTunnel.
- Example—Specifying "localQosProfiles.vlan" returns the name of the QoS profile in the VLAN layer.

loginName

- Subscriber's login name.
- Value—Login name
- Guidelines—The format of the login name varies. A loginName can be of form subscriber, domain\subscriber, subscriber@domain, or as otherwise defined by the login setup of the manager.
- Example—idp@idp

nasIp

- IP address of the router.
- Value—String

nasPort

- Numeric identifier that the router uses to identify the interface to RADIUS.
- Value—Integer

portId

- Port identifier of an interface.
- Value—Includes interface name and additional layer 2 information
- Example—portId=" fastEthernet 3/1" (There is a space between fastEthernet and slot number 3/1 in the nasPort field.)

primaryUserName

- PPP login name or the public DHCP username.
- Value—Subscriber name
- Example—primaryUserName=" peter"

radiusClass

- RADIUS class attribute of the service definition.
- Value—String
- Example—radiusClass=" Premium"

serviceName

- Identifier of the service.

serviceScope

- Identifier of the service scope.

serviceSessionName

- Identifier of the service session.

serviceSessionTag

- Tag for the service session.

sspHost

- Name of host on which the SAE is installed.

substitutions.<substitution name>

- Substitution with the specified name passed in at service activation.

userIp

- IP address of the subscriber.
- Value—String

userMacAddress

- Media access control (MAC) address of the DHCP subscriber.
- Value—Valid MAC address
- Example—userMacAddress=" 00:11:22:33:44:55"

userType

- Type of subscriber.

vrName

- Name of virtual router.
- Value—Virtual router name in the format <virtualRouter>@<router>
- Example—vrName="default@e_series5"

Using Functions for Backbone Congestion Point Classification Scripts

SRC ACP provides the following functions to use in backbone congestion point classification scripts:

- **getNicProxy(name)**—Get the NIC proxy defined under the current SRC ACP configuration group.
 - name—The name of the NIC proxy as defined under the SRC ACP configuration group.
- **nicLookupSingle(name, nicKey, constraints)**—Perform a NIC lookup using the specified NIC key and constraints with the NIC proxy defined under the current SRC ACP shared configuration group. The NIC key must uniquely identify a NIC value. If more than one result matches the same key, this function will raise the `AmbiguousKeyException` exception.
 - name—Name of the NIC proxy.
 - nicKey—String used as key for NIC lookup.
 - constraints (optional)—Map of NIC constraint information associated with the NIC key.

This function returns the lookup result as (nicValue, intermediateValues), where intermediateValues is a map of the intermediate name and value pair.

- **nicLookup(name, nicKey, constraints)**—Perform a NIC lookup using the specified NIC key and constraints for the NIC proxy defined under the current SRC ACP shared configuration group.
 - name—Name of the NIC proxy.
 - nicKey—String used as key for NIC lookup.
 - constraints (optional)—Map of NIC constraint information associated with the NIC key.

This function returns the lookup result as an array of (nicValue, intermediateValues), where intermediateValues is a map of the intermediate name and value pair.

- `nicInvalidateLookup(name, nicKey, nicValue, constraints)`--Used to signal to a NIC proxy that a key/value pair (returned from one of the lookup methods) resulted in a failure when the value was used. If the NIC proxy has this result cached, it will be removed from the cache.
 - `name`—Name of the NIC proxy.
 - `nicKey`—A string used as NIC key that was passed to the previous lookup operation.
 - `nicValue`—The NIC value returned from the previous lookup operation.
 - `constraints(optional)`—Map of NIC constraint information associated with the NIC key.
- `slot(nasPortId)`—Collects the slot number from the `nasPortId` or `interfaceName`.
- `port(nasPortId)`—Collects the port number from the `nasPortId` or `interfaceName`.
- `l2id(nasPortId)`—Collects the layer 2 ID from the `nasPortId` (VLAN id or ATM vpi.vci).
- `escape(string)`—Replaces any slash with the escape sequence `\`.

Configuring Congestion Point Profiles in the Directory

If you are using congestion point classification, you do not need to configure congestion point profiles.

To configure individual backbone congestion point profiles:

1. From configuration mode, access the configuration statement that configures congestion point profiles.

```
user@host# edit shared congestion-points profile name
```

Enter the name of the virtual router that supports the congestion point.

2. (Optional) Verify your configuration.

```
[edit shared congestion-points profile name]
user@host# show
```

Assigning Interfaces to Congestion Point Profiles

If you are using congestion point classification, you do not need to assign interfaces to congestion point profiles.

You must assign interfaces either to VRs or to individual services under the VRs. Services inherit interface assignments from the associated VR unless you assign an interface to the individual service. This network interface lists the DNS of interfaces associated with backbone congestion point profiles.

Use the following configuration statements to configure interface assignments:

```
shared congestion-points profile name {  
  interface [interface...];  
}
```

To assign interfaces to congestion point profiles:

1. From configuration mode, access the configuration statement that configures congestion point profiles.

```
user@host# edit shared congestion-points profile name
```

Enter the name of the network device to which you want to assign the congestion point profile.

2. (Optional) Specify the interfaces associated with a congestion point profile for this subscriber.

```
[edit shared congestion-points profile name]  
user@host# set interface interface
```

3. (Optional) Verify your configuration.

```
[edit shared congestion-points profile name]  
user@host# show
```

Related Topics

- Configuring SRC-ACP to Manage the Backbone Network (C-Web Interface)
- Configuring SRC ACP to Manage the Edge Network (SRC CLI) on page 255
- Viewing Information About Services in the Backbone Network (SRC CLI) on page 285
- Overview of SRC ACP on page 223

Configuring Congestion Point Classification (SRC CLI)

- Overview of Congestion Point Classification on page 269
- Configuration Statements for Congestion Point Classification on page 270
- Classifying Congestion Points (SRC CLI) on page 270
- Defining a Congestion Point Profile (SRC CLI) on page 276
- Congestion Point Expressions on page 276

Overview of Congestion Point Classification

Congestion point classification allows you to automate and scale the configuration of congestion points. SRC ACP uses classification scripts to determine which congestion point to load for a subscriber. SRC ACP can select the congestion point from congestion point profiles or subscriber profiles.

Congestion Point Classification Scripts

The congestion point classification scripts consist of targets and criteria.

- A target is the result of the classification script. The result of congestion point classification scripts is an LDAP search string that is used to find a unique congestion point in the directory. If no classification scripts are configured, the result of congestion point classification scripts is an LDAP search string for the subscriber profile of the particular subscriber.
- Criteria are match criteria. The script attempts to match criteria in the script to information sent from the router. Match criteria for a congestion point classification script might be a subscriber distinguished name (DN) or an interface name.

Each script can have multiple targets, and each target can have multiple criteria. When an object needs classification, the script processes the targets in turn. Within each target, the script processes criteria sequentially. When it finds that the classification criteria for a target match, it returns the target to SRC ACP.

Because classification scripts examine criteria sequentially as the criteria appear in the script, you should put more specific criteria at the beginning of the script and less specific criteria at the end of the script.

Congestion Point Profiles

Congestion point profiles are used to share congestion points that are generated based on dynamic configuration information. SRC ACP uses congestion point profiles to determine the set of congestion points based on the classification script results.

Changes that you make to classification scripts do not affect subscriber sessions that are already established.

- Related Topics**
- Overview of SRC ACP on page 223
 - Classifying Congestion Points (SRC CLI) on page 270
 - Congestion Point Classification Criteria on page 272
 - Configuration Statements for Congestion Point Classification on page 270

Configuration Statements for Congestion Point Classification

Use the following configuration statements to configure congestion point classification at the **[edit]** hierarchy level.

```
shared acp congestion-point-classifier rule name {  
    target target;  
    script script;  
}  
shared acp congestion-point-classifier rule name condition name ...  
shared congestion-points congestion-point-profile name {  
    expression [expression...];  
}
```

For detailed information about each configuration statement, see the *SRC PE CLI Command Reference*.

- Related Topics**
- Classifying Congestion Points (SRC CLI) on page 270
 - Defining a Congestion Point Profile (SRC CLI) on page 276
 - Congestion Point Expressions on page 276
 - Overview of Congestion Point Classification on page 269

Classifying Congestion Points (SRC CLI)

The tasks to classify congestion points are:

1. Configuring Targets and Criteria for Classification Scripts on page 270
2. Configuring Classification Scripts Contents for Classification Scripts on page 271
3. Configuring Congestion Point Classification Targets on page 271

Configuring Targets and Criteria for Classification Scripts

To define a target and criteria for the congestion point classification script:

1. From configuration mode, access the configuration statement that configures congestion point scripts. In this sample procedure, the scripts are configured in the config group.

```
user@host# edit shared acp group config congestion-point-classifier rule name
```

Enter a name for the congestion point classification script.

2. Specify the target for the classification script.

```
[edit shared acp group config congestion-point-classifier rule name]
user@host# set target target
```

For information about classification targets, see “Classifying Congestion Points (SRC CLI)” on page 270.

3. Specify the classification criteria for the target.

```
[edit shared acp group config congestion-point-classifier rule name]
user@host# set condition condition
```

For information about classification criteria, see “Congestion Point Classification Criteria” on page 272.

Configuring Classification Scripts Contents for Classification Scripts

To use the contents of a classification script to another object for the congestion point classification script:

1. From configuration mode, access the configuration statement that configures congestion point scripts. In this sample procedure, the scripts are configured in the config group.

```
user@host# edit shared acp group config congestion-point-classifier rule name
```

Enter a name for the congestion point classification script.

2. Specify the classification script that you want to use.

```
[edit shared acp group config congestion-point-classifier rule name]
user@host# set script script
```

Configuring Congestion Point Classification Targets

The target of the congestion point classification script is an LDAP search string. The search string uses a syntax similar to an LDAP URL (see RFC 2255—The LDAP URL Format (December 1997)). The syntax is:

```
baseDN [ ? [ attributes ] [ ? [ scope ] [ ? [ filter ] ] ] ]
```

- baseDN—Distinguished name (DN) of the object where the LDAP search starts.
- attributes—Is ignored.
- scope—Scope of search in the directory:

- base—Default; searches the base DN only.
- one—Searches the direct children of the base DN.
- sub—Searches the complete subtree below the base DN.
- filter—An RFC 2254–style LDAP search filter expression; for example, (uniqueId=<-userName->). See RFC 2254—The String Representation of LDAP Search Filters (December 1997).

With the exception of baseDN all the fields are optional.

The result of the LDAP search must be exactly one directory object. If no object or more than one object is found, congestion points for the subscriber are not loaded and all service activations for the subscriber are denied.

Congestion Point Classification Criteria

Congestion point classification criteria define match criteria that are used to find the congestion point profile. Use the fields in this topic to define classification criteria.

accountingId

- Value of directory attribute accountingUserId.

authUserId

- Identifier that a subscriber uses for authentication.
- Value—Username

dhcpPacket

- Content of the DHCP discover request.
- Value—Byte array
 - First 4 octets—Gateway IP address (giaddr field)
 - Remaining octets—DHCP options

For more information, see RFC 2131—Dynamic Host Configuration Protocol (March 1997) and RFC 2132—DHCP Options and BOOTP Vendor Extensions (March 1997).

domain

- Name of the domain used for secondary authentication.
- Value—Valid domain name
- Example—domain="isp99.com"

ifRadiusClass

- RADIUS class attribute on the JUNOS interface.
- Value—RADIUS class name
- Example—ifRadiusClass=" acpe"

ifSessionId

- Identifier for RADIUS accounting on the JUNOS interface.

interfaceAlias

- Description of the interface.
- Value—Interface description that is configured on the JUNOS router with the **interface ip description** command
- Example—interfaceAlias=" dhcp-subscriber12"

interfaceDescr

- Alternate name for the interface that is used by SNMP. This name is a system-generated name.
- Value
 - On a JUNOS router, the format of the description is
ip<slot>/<port>.<subinterface>
 - On the JUNOS routing platform, interfaceDescr is the same as interfaceName.
- Example—interfaceDescr=" IP3/1"

interfaceName

- Name of the interface.
- Value
 - Name of the interface in your router CLI syntax
 - FORWARDING_INTERFACE for routing instance (used by traffic mirroring)
- Example—For JUNOS routers: interfaceName=" fastEthernet6/0"
For JUNOS routing platforms: interfaceName="fe-0/1/0.0"
For forwarding interface: interfaceName="FORWARDING_INTERFACE"

localQosProfiles.<layer name>

- Local QoS profile in the specified layer. Local QoS profiles refer to profiles that are attached using the JunosE router CLI or the Service Manager and not through a SAE.
- Value—String
 - The <layer name> is one of the following values: ip, ipv6, lac, svlan, vlan, ethernet, atmVp, atmVc, atm, bridge, frVc, ipTunnel, l2tpTunnel.
- Example—Specifying “localQosProfiles.vlan” returns the name of the QoS profile in the VLAN layer.

loginName

- Subscriber's login name.
- Value—Login name
- Guidelines—The format of the login name varies. A loginName can be of form subscriber, domain\subscriber, subscriber@domain, or as otherwise defined by the login setup of the manager.
- Example—idp@idp

nasIp

- IP address of the router.
- Value—Byte array
 - For IPv4 address—4 octets in network byte order
 - For IPv6 address—16 octets in network byte order

nasPort

- Numeric identifier that the router uses to identify the interface to RADIUS.
- Value—Integer

portId

- Port identifier of an interface.
- Value—Includes interface name and additional layer 2 information
- Example—portId=“ fastEthernet 3/1” (There is a space between fastEthernet and slot number 3/1 in the nasPort field.)

primaryUserName

- PPP login name or the public DHCP username.
- Value—Subscriber name
- Example—primaryUserName=“ peter”

radiusClass

- RADIUS class attribute of the service definition.
- Value—RADIUS class name
- Example—radiusClass=" Premium"

routerName

- Name of virtual router.
- Value—Virtual router name in the format <virtualRouter>@<router>
- Example—routerName=" default@e_series5"

sessionId

- Identifier of RADIUS session for the subscriber session.

serviceBundle

- Content of the RADIUS vendor-specific attribute for the service bundle.
- Value—Name of a service bundle
- Example—serviceBundle=" goldSubscriber"

sspHost

- Name of host on which the SAE is installed.

userDn

- DN of a subscriber in the directory.
- Value—DN of a subscriber profile

userIp

- IP address of the subscriber.
- Value—Byte array
 - For IPv4 address—4 octets in network byte order
 - For IPv6 address—16 octets in network byte order

userMacAddress

- Media access control (MAC) address of the DHCP subscriber.
- Value—Valid MAC address
- Example—userMacAddress=" 00:11:22:33:44:55"

userType

- Type of subscriber.

- Related Topics**
- Configuring Congestion Point Classification (C-Web Interface)
 - Configuration Statements for Congestion Point Classification on page 270
 - Viewing Congestion Point Information by DN (SRC CLI) on page 290
 - Congestion Point Expressions on page 276
 - Overview of Congestion Point Classification on page 269

Defining a Congestion Point Profile (SRC CLI)

You can create a congestion point profile that automatically performs congestion point classification. This profile supports only access network mode for SRC ACP.

Use the following configuration statements to configure congestion point profiles:

```
shared congestion-points congestion-point-profile name {  
    expression [expression...];  
}
```

To define a congestion point profile:

1. From configuration mode, access the configuration statement that configures congestion point profiles.

```
user@host# edit shared congestion-points congestion-point-profile name
```

Enter a name for the profile.

2. Specify congestion point expressions.

```
[edit shared congestion-points congestion-point-profile name]  
user@host# set expression [expression...]
```

For information about congestion point expressions, see “Congestion Point Expressions” on page 276.

- Related Topics**
- Defining a Congestion Point Profile (C-Web Interface)
 - Classifying Congestion Points (SRC CLI) on page 270
 - Configuration Statements for Congestion Point Classification on page 270
 - Overview of Congestion Point Classification on page 269

Congestion Point Expressions

You can enter a congestion point expression by using the syntax listed in this topic. You can also embed Python scripting expressions within the congestion point expression.

If you embed Python expressions within a congestion point expression, use the escape sequence `<- then ->` to enclose the Python expression. See “Methods for Use with Scripting Expressions” on page 277 and “Match Criteria for Congestion Point Classification” on page 278.

The syntax for a congestion point expression is:

`<NetworkDevice>/<NetworkInterface>[/<CongestionPoint>]`

- `<NetworkDevice>`—Network device listed in the directory.
- `<NetworkInterface>`—Network interface listed in the directory.

For information about interfaces, see Overview of Classification Scripts .

- `<CongestionPoint>`—(Optional) Name of an instance of a congestion point that is automatically created.

If one of the elements with the path contains a slash (/), use a backslash (\) as an escape character for the slash. For example, `\.`

Expressions in Templates for Congestion Point Profiles

You can create a congestion point profile to be used as a template for other profiles. Templates simplify management of congestion points. Rather than configuring each congestion point individually, you can create templates to define common parameters for a class of individual congestion points.

For example, in an environment in which VLAN interfaces GigabitEthernet1/0.1 through GigabitEthernet1/0.1000 have the same available bandwidth, you can specify the characteristics of the VLAN interface once and have SRC ACP create the congestion points based on the template configuration.

When a congestion point expression has the third element (`<CongestionPoint>`), SRC ACP uses the `<NetworkDevice>/<NetworkInterface>` part of the expression to load the congestion point from the directory, and uses it as a template to create a congestion point in memory for subscriber. The `<CongestionPoint>` part of the expression distinguishes each congestion point (available bandwidth) created from this template.

Methods for Use with Scripting Expressions

SRC ACP provides the following methods to use in scripting expressions:

- `slot(nasPortId)`—Collects the slot number from the `nasPortId` or `interfaceName`
Example—`slot(" atm 4/5:0.32") == " 4"`
- `port(nasPortId)`—Collects the port number from the `nasPortId` or `interfaceName`
Example—`port(" atm 4/5:0.32") == " 5"`
- `l2id(nasPortId)`—Collects the layer 2 ID from the `nasPortId` (VLAN id or ATM vpi.vci)
Example—`l2id(" atm 4/5:0.32") == " 0.32"`
- `escape(string)`—Replaces any slash with the escape sequence `\`

Example—`escape("atm 4/5") == "atm 4√5"`

Match Criteria for Congestion Point Classification

You can use the match criteria in Python scripting expressions for a congestion point expression. For more information about the match criteria, see “Congestion Point Classification Criteria” on page 272.

- Related Topics**
- Overview of Congestion Point Classification on page 269
 - Classifying Congestion Points (SRC CLI) on page 270
 - Defining a Congestion Point Profile (SRC CLI) on page 276
 - Configuration Statements for Congestion Point Classification on page 270

CHAPTER 20

Managing SRC ACP (SRC CLI)

Topics in this chapter include:

- Starting SRC ACP on page 279
- Stopping SRC ACP on page 279
- Reorganizing the File That Contains ACP Data on page 280
- Modifying Congestion Points on page 280

Starting SRC ACP

To start SRC ACP:

```
user@host> enable component acp
```

- Related Topics**
- Stopping SRC ACP on page 279
 - Configuring SRC ACP (SRC CLI) on page 236
 - Reorganizing the File That Contains ACP Data on page 280
 - Viewing General Statistics for SRC ACP (C-Web Interface) on page 308
 - Overview of SRC ACP on page 223

Stopping SRC ACP

To stop SRC ACP:

```
user@host> disable component acp
```

- Related Topics**
- Starting SRC ACP on page 279
 - Reorganizing the File That Contains ACP Data on page 280
 - Viewing General Statistics for SRC ACP (C-Web Interface) on page 308
 - Overview of SRC ACP on page 223

Reorganizing the File That Contains ACP Data

Periodically, you should reorganize the files that contain ACP data about subscribers, services, and congestion points. This action reduces the sizes of these files. To do so:

```
user@host> request acp reorganize-backup-database
```

Related Topics

- Stopping SRC ACP on page 279
- Starting SRC ACP on page 279
- Viewing General Statistics for SRC ACP (C-Web Interface) on page 308
- Overview of SRC ACP on page 223

Modifying Congestion Points

By default, SRC ACP does not register changes in congestion points until you stop and restart SRC ACP. To modify the congestion point associated with a service without stopping and starting SRC ACP:

1. Make sure that no subscribers have subscriptions to services that use the congestion point you want to modify.
2. From configuration mode, access the configuration statement that configures SRC ACP connections to the services' directory.

```
user@host# edit shared acp configuration ldap service-data
```

3. Specify whether SRC ACP detects changes in the backbone congestion point for a service while SRC ACP is operative.

```
[edit shared acp configuration ldap service-data]  
user@host# set reload-congestion-points
```

4. Wait for 30 seconds before you proceed to the next step.

Depending on the value of the polling interval for directory eventing, SRC ACP may take up to 30 seconds to register the change to the **reload-congestion-points** option. If you modify the congestion point before SRC ACP registers the new setting for the **reload-congestion-points** option, SRC ACP will not register the change for the congestion point.

5. Modify the congestion point in the service definition.
SRC ACP immediately registers the change.
6. From configuration mode, access the configuration statement that configures SRC ACP connections to the services' directory.

```
user@host# edit shared acp configuration ldap service-data
```


7. Specify whether SRC ACP detects changes in the backbone congestion point for a service while SRC ACP is operative.

```
[edit shared acp configuration ldap service-data]  
user@host# set reload-congestion-points
```


Monitoring Admission Control (SRC CLI)

- Viewing Information About Subscriber Sessions in the Edge Network (SRC CLI) on page 283
- Viewing Edge Congestion Point Information by DN (SRC CLI) on page 284
- Viewing Edge Congestion Point Information by Subscriber Session (SRC CLI) on page 285
- Viewing Information About Services in the Backbone Network (SRC CLI) on page 285
- Viewing Backbone Congestion Point Information by DN (SRC CLI) on page 286
- Viewing Backbone Congestion Point Information by Service (SRC CLI) on page 287
- Viewing Action Congestion Point Information by Service (SRC CLI) on page 287
- Viewing Action Congestion Point Information by Congestion Point (SRC CLI) on page 288
- Viewing Information About Subscribers Obtained from External Applications (SRC CLI) on page 289
- Viewing Congestion Point Information by DN (SRC CLI) on page 290
- Viewing Congestion Point Information by Name (SRC CLI) on page 290
- Viewing SNMP Information for Devices (SRC CLI) on page 291
- Viewing SNMP Information for the Directory (SRC CLI) on page 291
- Viewing SNMP Information for SRC ACP (SRC CLI) on page 292

Viewing Information About Subscriber Sessions in the Edge Network (SRC CLI)

Purpose Display information about the subscriber session.

Action To display information about the current subscriber sessions in memory:

```
user@host> show acp edge subscriber
```

To display information about specific subscriber sessions:

```
user@host> show acp edge subscriber session-id session-id
```

Enter all or part of the subscriber session ID to list all matching subscriber sessions.

To display information about the subscriber sessions from a specific virtual router:

```
user@host> show acp edge subscriber virtual-router-name virtual-router-name
```

Enter a virtual router name to list subscriber sessions from a particular virtual router.

To display subscriber session attributes for the current subscriber sessions:

```
user@host> show acp edge subscriber brief
```

By default, information about the subscriber session attributes, service sessions, and associated congestion points is displayed.

- Related Topics**
- Configuring SRC ACP to Manage the Edge Network (SRC CLI) on page 255
 - Viewing Information About Subscriber Sessions in the Edge Network (C-Web Interface) on page 293
 - Viewing Information About Subscribers Obtained from External Applications (SRC CLI) on page 289

Viewing Edge Congestion Point Information by DN (SRC CLI)

Purpose View edge congestion point information by DN.

Action To display information about edge congestion points by DN:

```
user@host> show acp edge congestion-point dn
```

To display information about specific congestion points by DN:

```
user@host> show acp edge congestion-point dn congestion-point-dn  
congestion-point-dn
```

Enter a partial congestion point DN to list all matching congestion points.

To display information about specific congestion points that were generated dynamically by instance ID:

```
user@host> show acp edge congestion-point dn instance-id instance-id  
user@host> show acp edge congestion-point dn congestion-point-dn  
congestion-point-dn instance-id instance-id
```

When a congestion point is dynamically generated with a congestion point profile, the generated instance ID is appended to the congestion point DN. Enter a partial instance ID to list all matching congestion points.

To display information about the congestion points from a specific virtual router:

```
user@host> show acp edge congestion-point dn virtual-router-name  
virtual-router-name
```

Enter a virtual router name to list congestion points from a particular virtual router.

To display congestion point DNs:

```
user@host> show acp edge congestion-point dn brief
```

By default, information about the congestion point attributes and congestion point bandwidth usage is displayed.

To restrict the number of displayed results:

```
user@host> show acp edge congestion-point dn maximum-results maximum-results
```

- Related Topics**
- Configuring SRC ACP to Manage the Edge Network (SRC CLI) on page 255
 - Viewing Information About Edge Congestion Points by DN (C-Web Interface) on page 294
 - Viewing Edge Congestion Point Information by Subscriber Session (SRC CLI) on page 285

Viewing Edge Congestion Point Information by Subscriber Session (SRC CLI)

Purpose View edge congestion point information by subscriber session.

Action To display information about edge congestion points by subscriber session:

```
user@host> show acp edge congestion-point subscriber-session-id
```

To display information about specific congestion points by subscriber session:

```
user@host> show acp edge congestion-point subscriber-session-id session-id
session-id
```

Enter a partial subscriber session ID to list all matching congestion points.

To display information about the congestion points from a specific virtual router:

```
user@host> show acp edge congestion-point subscriber-session-id
virtual-router-name virtual-router-name
```

Enter a virtual router name to list congestion points from a particular virtual router.

To display congestion point DNs:

```
user@host> show acp edge congestion-point subscriber-session-id brief
```

By default, information about the congestion point attributes and congestion point bandwidth is displayed.

To restrict the number of displayed results:

```
user@host> show acp edge congestion-point subscriber-session-id maximum-results
maximum-results
```

- Related Topics**
- Configuring SRC ACP to Manage the Edge Network (SRC CLI) on page 255
 - Viewing Information About Edge Congestion Points by Subscriber Session (C-Web Interface) on page 295
 - Viewing Edge Congestion Point Information by DN (SRC CLI) on page 284

Viewing Information About Services in the Backbone Network (SRC CLI)

Purpose View information about services in the backbone network.

Action To display information about services that SRC ACP manages in the backbone network:

```
user@host> show acp backbone service
```

To display information about specific backbone service used to generate congestion points:

```
user@host> show acp backbone service service-name service-name
```

Enter a partial service name to list all matching backbone services.

To display information about the backbone services from a specific virtual router:

```
user@host> show acp backbone service virtual-router-name virtual-router-name
```

Enter a virtual router name to list backbone services from a particular virtual router.

To display backbone service attributes:

```
user@host> show acp backbone service brief
```

By default, information about the backbone service attributes, service sessions, and associated congestion points is displayed.

- Related Topics**
- Configuring SRC ACP to Manage the Backbone Network (SRC CLI) on page 259
 - Viewing Information About Services in a Backbone Network (C-Web Interface) on page 296
 - Viewing Backbone Congestion Point Information by Service (SRC CLI) on page 287

Viewing Backbone Congestion Point Information by DN (SRC CLI)

Purpose View backbone congestion point information by DN.

Action To display information about backbone congestion points by DN:

```
user@host> show acp backbone congestion-point dn
```

To display information about specific congestion points by DN:

```
user@host> show acp backbone congestion-point dn congestion-point-dn  
congestion-point-dn
```

Enter a partial congestion point DN to list all matching congestion points.

To display information about the congestion points from a specific virtual router:

```
user@host> show acp backbone congestion-point dn virtual-router-name  
virtual-router-name
```

Enter a virtual router name to list congestion points from a particular virtual router.

To display congestion point DNs:

```
user@host> show acp backbone congestion-point dn brief
```

By default, information about the congestion point attributes and congestion point bandwidth usage is displayed.

- Related Topics**
- Configuring SRC ACP to Manage the Backbone Network (SRC CLI) on page 259
 - Viewing Information About Congestion Points in a Backbone Network by DN (C-Web Interface) on page 299
 - Viewing Backbone Congestion Point Information by Service (SRC CLI) on page 287

Viewing Backbone Congestion Point Information by Service (SRC CLI)

Purpose View backbone congestion point information by service.

Action To display information about backbone congestion points by service:

```
user@host> show acp backbone congestion-point congestion-point-expression
```

To display information about specific backbone services used to generate congestion points:

```
user@host> show acp backbone congestion-point congestion-point-expression
service-name service-name
```

Enter a partial service name to list all matching backbone services.

To display information about the backbone services from a specific virtual router:

```
user@host> show acp backbone congestion-point congestion-point-expression
virtual-router-name virtual-router-name
```

Enter a virtual router name to list backbone services from a particular virtual router.

To display congestion point DNs:

```
user@host> show acp backbone congestion-point congestion-point-expression brief
```

By default, information about the congestion point attributes and congestion point bandwidth is displayed.

- Related Topics**
- Configuring SRC ACP to Manage the Backbone Network (SRC CLI) on page 259
 - Viewing Information About Congestion Points in a Backbone Network by Expression (C-Web Interface) on page 298
 - Viewing Backbone Congestion Point Information by DN (SRC CLI) on page 286

Viewing Action Congestion Point Information by Service (SRC CLI)

Purpose View action congestion point information by service.

Action To display information about services that SRC ACP manages in the backbone network:

```
user@host> show acp backbone service
```

To display information about specific backbone services used to generate congestion points:

```
user@host> show acp backbone service service-name service-name
```

Enter a partial service name to list all matching backbone services.

To display information about the backbone services from a specific virtual router:

```
user@host> show acp backbone service virtual-router-name virtual-router-name
```

To display backbone service attributes:

```
user@host> show acp backbone service brief
```

By default, information about the backbone service attributes, service sessions, and associated congestion points is displayed.

- Related Topics**
- Configuring SRC ACP to Manage the Backbone Network (SRC CLI) on page 259
 - Viewing Information about Action Congestion Points in a Backbone Network by Service (C-Web Interface) on page 300
 - Viewing Action Congestion Point Information by Congestion Point (SRC CLI) on page 288

Viewing Action Congestion Point Information by Congestion Point (SRC CLI)

Purpose View action congestion point information by congestion point.

Action To display information about backbone congestion points by service:

```
user@host> show acp backbone congestion-point congestion-point-expression
```

To display information about specific backbone services used to generate congestion points:

```
user@host> show acp backbone congestion-point congestion-point-expression  
service-name service-name
```

Enter a partial service name to list all matching backbone services.

To display information about the backbone services from a specific virtual router:

```
user@host> show acp backbone congestion-point congestion-point-expression  
virtual-router-name virtual-router-name
```

Enter a virtual router name to list backbone services from a particular virtual router.

To display information about the backbone services from a specific interface:

```
user@host> show acp backbone congestion-point congestion-point-expression  
interface-name interface-name
```

Enter an interface name to list backbone services from a particular interface.

To display information about the backbone services for a specific interface description:

```
user@host> show acp backbone congestion-point congestion-point-expression  
interface-description interface-description
```

Enter an interface description to list backbone services for a particular description.

To display information about the backbone services from a specific interface alias:

```
user@host> show acp backbone congestion-point congestion-point-expression  
interface-alias interface-alias
```

Enter an interface alias to list backbone services from a particular alias.

To display information about the backbone services for a specific NAS port ID:

```
user@host> show acp backbone congestion-point congestion-point-expression  
nasPort-id nasPort-id
```

Enter a NAS port ID to list backbone services from a particular ID.

To display congestion point DNs:

```
user@host> show acp backbone congestion-point congestion-point-expression brief
```

By default, information about the congestion point attributes and congestion point bandwidth is displayed.

- Related Topics**
- Configuring SRC ACP to Manage the Backbone Network (SRC CLI) on page 259
 - Viewing Backbone Congestion Point Information by DN (SRC CLI) on page 286
 - Viewing Backbone Congestion Point Information by Service (SRC CLI) on page 287
 - Viewing Action Congestion Point Information by Service (SRC CLI) on page 287

Viewing Information About Subscribers Obtained from External Applications (SRC CLI)

Purpose View information about subscribers obtained from external applications.

Action To display information about subscribers added through an external application:

```
user@host> show acp remote-update subscriber
```

To display information about subscribers connected from a specific device:

```
user@host> show acp remote-update subscriber device-name device-name
```

Enter a device name to list subscribers connected from a particular device.

To display information about specific subscribers connected from a specific interface:

```
user@host> show acp remote-update subscriber nas-port-id nas-port-id
```

Enter the NAS port ID of interface to list all matching subscribers connected from a particular interface.

To display information about specific subscribers connected from a specific NAS IP address:

```
user@host> show acp remote-update subscriber nas-ip nas-ip
```

Enter the NAS IP address of the device to list all matching subscribers connected from a particular device.

To display information about specific subscribers connected from a specific subscriber IP address:

```
user@host> show acp remote-update subscriber subscriber-ip subscriber-ip
```

Enter the subscriber IP address to list all matching subscribers connected from a particular address.

To display information about the subscribers from a specific phone number:

```
user@host> show acp remote-update subscriber phone phone
```

Enter a phone number to list subscribers from a particular phone number.

To display subscriber attributes:

```
user@host> show acp remote-update subscriber brief
```

By default, information about the subscriber attributes, service sessions, and associated congestion points is displayed.

- Related Topics**
- Viewing Information About Subscribers Obtained from External Applications (C-Web Interface) on page 304
 - Viewing Congestion Point Information by DN (SRC CLI) on page 290
 - Viewing Congestion Point Information by Name (SRC CLI) on page 290

Viewing Congestion Point Information by DN (SRC CLI)

Purpose View congestion point information by DN.

Action To display information about congestion points added through an external application by DN:

```
user@host> show acp remote-update congestion-point dn
```

To display information about specific congestion points by DN:

```
user@host> show acp remote-update congestion-point dn congestion-point-dn  
congestion-point-dn
```

Enter a partial congestion point DN to list all matching congestion points.

To display congestion point DNs:

```
user@host> show acp remote-update congestion-point dn brief
```

By default, information about the congestion point attributes and congestion point bandwidth usage is displayed.

- Related Topics**
- Viewing Information About Congestion Points from an External Application by DN (C-Web Interface) on page 305
 - Viewing Information About Subscribers Obtained from External Applications (SRC CLI) on page 289
 - Viewing Congestion Point Information by Name (SRC CLI) on page 290

Viewing Congestion Point Information by Name (SRC CLI)

Purpose View congestion point information by name.

Action To display information about congestion points added through an external application by interface name:

```
user@host> show acp remote-update congestion-point name
```

To display information about congestion points connected from a specific device:

```
user@host> show acp remote-update congestion-point name device-name device-name
```

Enter a device name to list congestion points connected from a particular device.

To display information about specific subscribers connected from a specific interface:

```
user@host> show acp remote-update congestion-point name interface-name
interface-name
```

Enter the interface name to list all matching congestion points connected from a particular interface.

To display congestion point DN:

```
user@host> show acp remote-update congestion-point name brief
```

By default, information about the congestion point attributes and congestion point bandwidth usage is displayed.

- Related Topics**
- Viewing Information About Congestion Points from an External Application by Interface Name (C-Web Interface) on page 306
 - Viewing Information About Subscribers Obtained from External Applications (SRC CLI) on page 289
 - Viewing Congestion Point Information by DN (SRC CLI) on page 290

Viewing SNMP Information for Devices (SRC CLI)

Purpose View SNMP information for devices.

Action To display statistics for SNMP information about each device:

```
user@host> show acp statistics device
```

To display statistics for SNMP information about specific devices:

```
user@host> show acp statistics device filter filter
```

Enter a partial device name to list information for all matching devices.

- Related Topics**
- Viewing SNMP Information for the Directory (SRC CLI) on page 291
 - Viewing SNMP Information for SRC ACP (SRC CLI) on page 292

Viewing SNMP Information for the Directory (SRC CLI)

Purpose View SNMP information for the directory.

Action To display statistics for directory SNMP information:

```
user@host> show acp statistics directory
```

- Related Topics**
- Viewing SNMP Information for Devices (SRC CLI) on page 291
 - Viewing SNMP Information for SRC ACP (SRC CLI) on page 292

Viewing SNMP Information for SRC ACP (SRC CLI)

Purpose View SNMP information for SRC ACP.

Action To display statistics for SRC ACP SNMP information:

```
user@host> show acp statistics general
```

- Related Topics**
- Viewing SNMP Information for Devices (SRC CLI) on page 291
 - Viewing SNMP Information for the Directory (SRC CLI) on page 291

CHAPTER 22

Monitoring Admission Control (C-Web Interface)

- Viewing Information About Subscriber Sessions in the Edge Network (C-Web Interface) on page 293
- Viewing Information About Edge Congestion Points by DN (C-Web Interface) on page 294
- Viewing Information About Edge Congestion Points by Subscriber Session (C-Web Interface) on page 295
- Viewing Information About Services in a Backbone Network (C-Web Interface) on page 296
- Viewing Information About Congestion Points in a Backbone Network by Expression (C-Web Interface) on page 298
- Viewing Information About Congestion Points in a Backbone Network by DN (C-Web Interface) on page 299
- Viewing Information about Action Congestion Points in a Backbone Network by Service (C-Web Interface) on page 300
- Viewing Information about Action Congestion Points in a Backbone Network by Expression (C-Web Interface) on page 302
- Viewing Information About Subscribers Obtained from External Applications (C-Web Interface) on page 304
- Viewing Information About Congestion Points from an External Application by DN (C-Web Interface) on page 305
- Viewing Information About Congestion Points from an External Application by Interface Name (C-Web Interface) on page 306
- Viewing Statistics for the SRC ACP Configuration (C-Web Interface) on page 307

Viewing Information About Subscriber Sessions in the Edge Network (C-Web Interface)

Purpose View information about subscriber sessions in the edge network with the C-Web interface.

Action To view information about subscriber sessions:

1. Click **ACP>Edge>Subscriber**.

The Edge/Subscriber pane appears.

Field	Description	Value	Default
Session Id	Subscriber session ID for which you want to list all matching subscriber sessions.	All or part of the subscriber session ID.	No value
Slot	Number of the slot for which you want to configure values.	Currently, the chassis has only one slot. The valid value is 0.	0
Style	Output style.	Choices: brief: Minimal information	detail
Virtual Router Name	Name of virtual router from which to list subscriber sessions.	Virtual router name	No value

2. In the Session ID box, enter a full or partial session ID name to display information about one or more specific sessions, or leave this field empty to display information about all sessions.
3. In the Slot box, enter the number of the slot for which you want to display subscriber session information.
4. Select an output style from the Style list.
5. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
6. Click **OK**.

The Edge/Subscriber pane displays a list of current subscriber sessions.

Related Topics

- Configuring SRC ACP to Manage the Edge Network (C-Web Interface)
- Viewing Information About Subscriber Sessions in the Edge Network (SRC CLI) on page 283
- Viewing Information About Edge Congestion Points by Subscriber Session (C-Web Interface) on page 295
- Viewing Information About Edge Congestion Points by DN (C-Web Interface) on page 294

Viewing Information About Edge Congestion Points by DN (C-Web Interface)

Purpose View information about edge congestion points by DN.

Action To view information about edge congestion points:

1. Click **ACP>Edge>Congestion Point>DN**.

The Edge/Congestion Point/DN pane appears.

Field	Description	Value	Default
Congestion Point Dn	DN of congestion point for which you want to list all matching congestion points.	All or part of the congestion point DN.	No value
Slot	Number of the slot for which you want to configure values.	Currently, the chassis has only one slot. The valid value is 0.	0
Style	Output style.	brief: Display congestion point DN	detail
Virtual Router Name	Name of virtual router from which to list congestion points.	Virtual router name	No value

2. In the Congestion Point DN box, enter a congestion point DN, or leave the box blank to view information for all DNs.
3. In the Slot box, enter the number of the slot for which you want to display congestion point information.
4. Select an output style from the Style list.
5. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
6. Click **OK**.

The Edge/Congestion Point/DN pane displays a list of congestion points.

Related Topics

- Configuring SRC ACP to Manage the Edge Network (C-Web Interface)
- Viewing Information About Edge Congestion Points by DN (C-Web Interface) on page 294
- Viewing Information About Edge Congestion Points by Subscriber Session (C-Web Interface) on page 295
- Viewing Information About Subscriber Sessions in the Edge Network (C-Web Interface) on page 293

Viewing Information About Edge Congestion Points by Subscriber Session (C-Web Interface)

Purpose View information about edge congestion points by subscriber session.

Action To view information about edge congestion points:

1. Click **ACP>Edge>Congestion Point>Subscriber Session ID**.

The Edge/Congestion Point/Subscriber Session ID pane appears.

Field	Description	Value	Default
Session Id	Subscriber session ID for which you want to list all matching congestion points.		No value
Slot	Number of the slot for which you want to configure values.		0
Style	Output style.		detail
Virtual Router Name	Name of virtual router from which to list congestion points.		No value

2. In the Session ID box, enter a full or partial session ID name to display information about one or more specific sessions, or leave the box empty to display information about all sessions.
3. In the Slot box, enter the number of the slot for which you want to display congestion point information.
4. Select an output style from the Style list.
5. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
6. Click **OK**.

The Edge/Congestion Point/Subscriber Session ID pane displays a list of congestion points.

Related Topics

- Configuring SRC ACP to Manage the Edge Network (C-Web Interface)
- Viewing Information About Edge Congestion Points by Subscriber Session (C-Web Interface) on page 295
- Viewing Information About Edge Congestion Points by DN (C-Web Interface) on page 294
- Viewing Information About Subscriber Sessions in the Edge Network (C-Web Interface) on page 293

Viewing Information About Services in a Backbone Network (C-Web Interface)

Purpose View information about services in a backbone network with the C-Web interface.

Action To view information about services in a backbone network:

1. Click **ACP>Backbone>Service**.

The Backbone/Service pane appears.

Monitor	Configure	Diagnose	Manage	Logged in as: admin	Refresh	Preferences	About	Logout
ACP	ACP							
CLI	Backbone / Service							
Component								
Date	Interface Alias	<input type="text"/>	Interface alias used by backbone service to generate congestion points. <i>Value:</i> Interface alias <i>Default:</i> No value					
Disk	Interface Description	<input type="text"/>	Description of interface used by backbone service to generate congestion points. <i>Value:</i> Interface description <i>Default:</i> No value					
Interfaces...	Interface Name	<input type="text"/>	Name of interface related to congestion points. <i>Value:</i> Interface name <i>Default:</i> No value					
Iptables...	Nas Port Id	<input type="text"/>	Interface NAS port ID used by backbone service to generate congestion points. <i>Value:</i> NAS port ID <i>Default:</i> No value					
JPS	Service Name	<input type="text"/>	Name of service used by backbone service to generate congestion points. <i>Value:</i> Service name <i>Default:</i> No value					
NIC	Slot	<input type="text"/>	Number of the slot for which you want to configure values. <i>Value:</i> Currently, the chassis has only one slot. The valid value is 0. <i>Default value:</i> 0					
NTP	Style	<input type="text" value="detail"/>	Output style. <i>Choices:</i> brief: Display backbone service attributes <i>Default value:</i> detail					
Redirect Server	Virtual Router Name	<input type="text"/>	Name of virtual router from which to list backbone services. <i>Value:</i> Virtual router name <i>Default:</i> No value					
Route...	<input type="button" value="OK"/> <input type="button" value="Reset"/>							

2. In the Interface Alias box, enter the interface alias used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
3. In the Interface Description box, enter the interface description used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
4. In the Interface Name box, enter the name of an interface to display information about one interface, or leave the box empty to display information about all interfaces.
5. In the NAS Port ID box, enter the NAS port ID used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
6. In the Service Name box, enter the name of a service to display information about one service, or leave the box empty to display information about all services.
7. In the Slot box, enter the number of the slot for which you want to display congestion point information.
8. Select an output style from the Style list.
9. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
10. Click **OK**.

The Backbone/Service pane displays a list of services.

For more information about viewing service information for action congestion points, see “Viewing Information about Action Congestion Points in a Backbone Network by Service (C-Web Interface)” on page 300 .

- Related Topics**
- Configuring SRC ACP to Manage the Backbone Network (SRC CLI) on page 259
 - Viewing Information About Services in the Backbone Network (SRC CLI) on page 285
 - Viewing Information About Congestion Points in a Backbone Network by Expression (C-Web Interface) on page 298
 - Viewing Information About Congestion Points in a Backbone Network by DN (C-Web Interface) on page 299

Viewing Information About Congestion Points in a Backbone Network by Expression (C-Web Interface)

Purpose View information about congestion points in a backbone network by expression.

Action To view information about congestion points by expression:

1. Click **ACP>Backbone>Congestion Point>Congestion Point Expression**.

The Backbone/Congestion Point/Congestion Point Expression pane appears.

Field	Description	Value	Default
Interface Alias	Interface alias used by backbone service to generate congestion points.	<input type="text"/>	Interface alias No value
Interface Description	Description of interface used by backbone service to generate congestion points.	<input type="text"/>	Interface description No value
Interface Name	Name of interface related to congestion points.	<input type="text"/>	Interface name No value
Nas Port Id	Interface NAS port ID used by backbone service to generate congestion points.	<input type="text"/>	NAS port ID No value
Service Name	Name of service used by backbone service to generate congestion points.	<input type="text"/>	Service name No value
Slot	Number of the slot for which you want to configure values.	<input type="text"/>	Currently, the chassis has only one slot. The valid value is 0. Default value: 0
Style	Output style.	<input type="text"/>	Choices: brief: Display congestion point attributes Default value: detail
Virtual Router Name	Name of virtual router from which to list congestion points.	<input type="text"/>	Virtual router name No value

OK Reset

2. In the Interface Alias box, enter the interface alias used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.

3. In the Interface Description box, enter the interface description used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
4. In the Interface Name box, enter the name of an interface to display information about one interface, or leave the box empty to display information about all interfaces.
5. In the NAS Port ID box, enter the NAS port ID used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
6. In the Service Name box, enter the name of a service to display information about one service, or leave the box empty to display information about all services.
7. In the Slot box, enter the number of the slot for which you want to display congestion point information.
8. Select an output style from the Style list.
9. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
10. Click **OK**.

The Backbone/Congestion Point/Congestion Point Expression pane displays a list of congestion points.

For more information about viewing information for action congestion points by expression, see “Viewing Information about Action Congestion Points in a Backbone Network by Expression (C-Web Interface)” on page 302.

- Related Topics**
- Configuring Congestion Points in the Directory
 - Viewing Information About Services in a Backbone Network (C-Web Interface) on page 296
 - Viewing Information About Congestion Points in a Backbone Network by DN (C-Web Interface) on page 299
 - Viewing Information about Action Congestion Points in a Backbone Network by Service (C-Web Interface) on page 300

Viewing Information About Congestion Points in a Backbone Network by DN (C-Web Interface)

Purpose View information about congestion points in a backbone network by DN.

Action To view information about congestion points by DN:

1. Click **ACP>Backbone>Congestion Point>DN**.

The Backbone/Congestion Point/DN pane appears.

ACP		Backbone / Congestion Point / DN	
Congestion Point Dn	<input type="text"/>	DN of congestion point for which you want to list all matching congestion points. <i>Value:</i> All or part of the congestion point DN. <i>Default:</i> No value	
Slot	<input type="text"/>	Number of the slot for which you want to configure values. <i>Value:</i> Currently, the chassis has only one slot. The valid value is 0. <i>Default value:</i> 0	
Style	<input type="text"/>	Output style. <i>Choices:</i> brief: Display congestion point DN <i>Default value:</i> detail	
Virtual Router Name	<input type="text"/>	Name of virtual router from which to list congestion points. <i>Value:</i> Virtual router name <i>Default:</i> No value	
<input type="button" value="OK"/> <input type="button" value="Reset"/>			

Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice, Privacy. Juniper Your Net.

2. In the Congestion Point DN box, enter a full or partial congestion point name to display information about one or more specific congestion points, or leave the box empty to display information about all congestion points.
3. In the Slot box, enter the number of the slot for which you want to display congestion point information.
4. Select an output style from the Style list.
5. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
6. Click **OK**.

The Backbone/Congestion Point/DN pane displays a list of congestion points.

Related Topics

- Configuring Congestion Points in the Directory
- Viewing Information About Services in the Backbone Network (SRC CLI) on page 285
- Viewing Information About Congestion Points in a Backbone Network by Expression (C-Web Interface) on page 298
- Viewing Information about Action Congestion Points in a Backbone Network by Service (C-Web Interface) on page 300

Viewing Information about Action Congestion Points in a Backbone Network by Service (C-Web Interface)

Purpose View information about action congestion points in a backbone network by service.

Action To view information about action congestion points in a backbone network by service:

1. Click **ACP>Backbone>Service**.

The Backbone/Service pane appears.

Monitor	Configure	Diagnose	Manage	Logged in as: admin	Refresh	Preferences	About	Logout
ACP	ACP							
CLI	Backbone / Service							
Component								
Date								
Disk								
Interfaces...								
Iptables...								
JPS								
NIC								
NTP								
Redirect Server								
Route...								
SAE								
Security								
System								
	Interface Alias	<input type="text"/>	Interface alias used by backbone service to generate congestion points. <i>Value:</i> Interface alias <i>Default:</i> No value					
	Interface Description	<input type="text"/>	Description of interface used by backbone service to generate congestion points. <i>Value:</i> Interface description <i>Default:</i> No value					
	Interface Name	<input type="text"/>	Name of interface related to congestion points. <i>Value:</i> Interface name <i>Default:</i> No value					
	Nas Port Id	<input type="text"/>	Interface NAS port ID used by backbone service to generate congestion points. <i>Value:</i> NAS port ID <i>Default:</i> No value					
	Service Name	<input type="text"/>	Name of service used by backbone service to generate congestion points. <i>Value:</i> Service name <i>Default:</i> No value					
	Slot	<input type="text"/>	Number of the slot for which you want to configure values. <i>Value:</i> Currently, the chassis has only one slot. The valid value is 0. <i>Default value:</i> 0					
	Style	<input type="text" value="detail"/>	Output style. <i>Choices:</i> brief: Display backbone service attributes <i>Default value:</i> detail					
	Virtual Router Name	<input type="text"/>	Name of virtual router from which to list backbone services. <i>Value:</i> Virtual router name <i>Default:</i> No value					
		<input type="button" value="OK"/> <input type="button" value="Reset"/>						

2. In the Interface Alias box, enter the interface alias used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
3. In the Interface Description box, enter the interface description used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
4. In the Interface Name box, enter the name of an interface to display information about one interface related to congestion points, or leave the box empty to display information about all interfaces.
5. In the NAS Port ID box, enter the NAS port ID used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
6. In the Service Name box, enter the name of a service to display information about one service, or leave the box empty to display information about all services.
7. In the Slot box, enter the number of the slot for which you want to display congestion point information.
8. Select an output style from the Style list.

9. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.

10. Click **OK**.

The Backbone/Service pane displays a list of congestion points.

- Related Topics**
- [Configuring Action Congestion Points](#)
 - [Viewing Information About Services in the Backbone Network \(SRC CLI\) on page 285](#)
 - [Viewing Information About Congestion Points in a Backbone Network by Expression \(C-Web Interface\) on page 298](#)
 - [Viewing Information About Congestion Points in a Backbone Network by DN \(C-Web Interface\) on page 299](#)

Viewing Information about Action Congestion Points in a Backbone Network by Expression (C-Web Interface)

Purpose View information about action congestion points in a backbone network by expression.

Action To view information about action congestion points in a backbone network by expression:

1. Click **ACP>Backbone>Congestion Point>Congestion Point Expression**.

The Backbone/Congestion Point/Congestion Point Expression pane appears.

Monitor	Configure	Diagnose	Manage	Logged in as: admin	Refresh	Preferences	About	Logout
ACP	ACP							
CLI	Backbone / Congestion Point / Congestion Point Expression							
Component								
Date								
Disk								
Interfaces...								
Iptables...								
JPS								
NIC								
NTP								
Redirect Server								
Route...								
SAE								
Security								
System								
	Interface Alias	<input type="text"/>	Interface alias used by backbone service to generate congestion points. <i>Value:</i> Interface alias <i>Default:</i> No value					
	Interface Description	<input type="text"/>	Description of interface used by backbone service to generate congestion points. <i>Value:</i> Interface description <i>Default:</i> No value					
	Interface Name	<input type="text"/>	Name of interface related to congestion points. <i>Value:</i> Interface name <i>Default:</i> No value					
	Nas Port Id	<input type="text"/>	Interface NAS port ID used by backbone service to generate congestion points. <i>Value:</i> NAS port ID <i>Default:</i> No value					
	Service Name	<input type="text"/>	Name of service used by backbone service to generate congestion points. <i>Value:</i> Service name <i>Default:</i> No value					
	Slot	<input type="text"/>	Number of the slot for which you want to configure values. <i>Value:</i> Currently, the chassis has only one slot. The valid value is 0. <i>Default value:</i> 0					
	Style	<input type="text"/>	Output style. <i>Choices:</i> brief: Display congestion point attributes <i>Default value:</i> detail					
	Virtual Router Name	<input type="text"/>	Name of virtual router from which to list congestion points. <i>Value:</i> Virtual router name <i>Default:</i> No value					
	<input type="button" value="OK"/> <input type="button" value="Reset"/>							

2. In the Interface Alias box, enter the interface alias used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
3. In the Interface Description box, enter the interface description used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
4. In the Interface Name box, enter the name of an interface to display information about one interface related to congestion points, or leave the box empty to display information about all interfaces.
5. In the NAS Port ID box, enter the NAS port ID used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
6. In the Service Name box, enter the name of a service to display information about one service, or leave the box empty to display information about all services.
7. In the Slot box, enter the number of the slot for which you want to display congestion point information.
8. Select an output style from the Style list.
9. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
10. Click **OK**.

The Backbone/Congestion Point/Congestion Point Expression pane displays a list of congestion points.

- Related Topics**
- [Configuring Action Congestion Points](#)
 - [Viewing Information about Action Congestion Points in a Backbone Network by Service \(C-Web Interface\) on page 300](#)
 - [Viewing Information About Congestion Points in a Backbone Network by Expression \(C-Web Interface\) on page 298](#)
 - [Viewing Information About Congestion Points in a Backbone Network by DN \(C-Web Interface\) on page 299](#)

Viewing Information About Subscribers Obtained from External Applications (C-Web Interface)

Purpose View information about subscribers obtained from external applications with the C-Web interface.

Action To view information about subscribers obtained from external applications:

1. Click **ACP>Remote Update>Subscriber**.

The Remote Update/Subscriber pane appears.

Field	Description	Value	Default
Device Name	Device name connected to subscriber.	Device name	No value
Nas Ip	NAS IP address of device connected to subscriber.	IP address	No value
Nas Port Id	NAS port ID of interface connected to subscriber.	NAS port ID	No value
Phone	Subscriber phone number.	Phone number	No value
Slot	Number of the slot for which you want to configure values.	Currently, the chassis has only one slot. The valid value is 0.	0
Style	Output style.	brief: Display congestion point DN	detail
Subscriber Ip	Subscriber IP address.	IP address	No value

2. In the Device Name box, enter the device name of the congestion point, or leave the box blank to display information about all devices.
3. In the NAS IP box, enter the NAS IP address of the device connected to the subscriber, or leave the box empty to display information about all subscribers.

4. In the NAS Port ID box, enter the NAS port ID connected to the subscriber, or leave the box empty to display information about all subscribers.
5. In the Phone box, enter the phone number of the subscriber, or leave the box blank to display information about all subscribers.
6. In the Slot box, enter the number of the slot for which you want to display external subscriber information.
7. Select an output style from the Style list.
8. In the Subscriber IP box, enter the subscriber IP address, or leave the box empty to display information about all subscribers.
9. Click **OK**.

The Remote Update/Subscriber pane displays the congestion points.

- Related Topics**
- Viewing Information About Subscribers Obtained from External Applications (SRC CLI) on page 289
 - Viewing Information About Congestion Points from an External Application by DN (C-Web Interface) on page 305
 - Viewing Information About Congestion Points from an External Application by Interface Name (C-Web Interface) on page 306

Viewing Information About Congestion Points from an External Application by DN (C-Web Interface)

Purpose View information about congestion points from an external application by DN.

Action To view information about congestion points added through an external application by DN:

1. Click **ACP>Remote Update>Congestion Point>DN**.

The Remote Update/Congestion Point/DN pane appears.

ACP Remote Update / Congestion Point / DN	
Congestion Point Dn	DN of congestion point for which you want to list all matching congestion points. <i>Value:</i> All or part of the congestion point DN. <i>Default:</i> No value
Slot	Number of the slot for which you want to configure values. <i>Value:</i> Currently, the chassis has only one slot. The valid value is 0. <i>Default value:</i> 0
Style	Output style. <i>Choices:</i> brief: Display congestion point DN <i>Default value:</i> detail

OK Reset

Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy. Juniper Your Net.

2. In the Congestion Point DN box, enter the DN of the congestion point, or leave the box blank to display information about all devices.
3. In the Slot box, enter the number of the slot for which you want to display congestion point information.
4. Select an output style from the Style list.
5. Click **OK**.

The Remote Update/Congestion Point/DN pane displays the congestion points.

- Related Topics**
- Viewing Congestion Point Information by DN (SRC CLI) on page 290
 - Viewing Information About Subscribers Obtained from External Applications (C-Web Interface) on page 304
 - Viewing Information About Congestion Points from an External Application by Interface Name (C-Web Interface) on page 306

Viewing Information About Congestion Points from an External Application by Interface Name (C-Web Interface)

Purpose View information about congestion points from an external application by interface name.

Action 1. Click **ACP>Remote Update>Congestion Point>Name**.

The Remote Update/Congestion Point/Name pane appears.

Field	Description	Value	Default
Device Name	Device name of the congestion point.		No value
Interface Name	Interface name of the congestion point.		No value
Slot	Number of the slot for which you want to configure values.	0	0
Style	Output style.	detail	detail

- In the Device Name box, enter the device name of the congestion point, or leave the box blank to display information about all devices.
- In the Interface Name box, enter the interface name of the congestion point, or leave the box blank to display information about all interfaces.
- In the Slot box, enter the number of the slot for which you want to display congestion point information.
- Select an output style from the Style list.
- Click **OK**.

The Remote Update/Congestion Point/Name pane displays the congestion points.

- Related Topics**
- Viewing Information About Subscribers Obtained from External Applications (C-Web Interface) on page 304
 - Viewing Information About Congestion Points from an External Application by DN (C-Web Interface) on page 305
 - Viewing Congestion Point Information by Name (SRC CLI) on page 290

Viewing Statistics for the SRC ACP Configuration (C-Web Interface)

- Viewing General Statistics for SRC ACP (C-Web Interface) on page 308
- Viewing Statistics for the SRC ACP Directory (C-Web Interface) on page 308
- Viewing Device Statistics for SRC ACP (C-Web Interface) on page 309

Viewing General Statistics for SRC ACP (C-Web Interface)

Purpose View general statistics for SRC ACP.

Action To view general statistics for SRC ACP:

1. Click **ACP>Statistics>General**.

The Statistics/General pane appears.

2. In the Slot box, enter the number of the slot for which you want to display general statistics.
3. Click **OK**.

The Statistics/General pane displays general SRC ACP statistics.

- Related Topics**
- Configuring SRC ACP (C-Web Interface)
 - Viewing Statistics for the SRC ACP Directory (C-Web Interface) on page 308
 - Viewing Device Statistics for SRC ACP (C-Web Interface) on page 309

Viewing Statistics for the SRC ACP Directory (C-Web Interface)

Purpose View statistics for the SRC ACP directory.

Action To view statistics about the SRC ACP directory:

1. Click **ACP>Statistics>Directory**.

The Statistics/Directory pane appears.

Monitor Configure Diagnose Manage Logged in as: admin Refresh Preferences About Logout

ACP

CLI

Component

Date

Disk

Interfaces...

Iptables...

JPS

NIC

NTP

Redirect Server

Route...

SAE

Security

System

Statistics / Directory

Slot

Number of the slot for which you want to configure values.
Value: Currently, the chassis has only one slot. The valid value is 0.
Default value: 0

OK Reset

Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy. Juniper Your Net.

- In the Slot box, enter the number of the slot for which you want to display directory statistics.
- Click **OK**.

The Statistics/Directory pane displays statistics for the SRC ACP directory.

Related Topics

- Configuring Local Properties for SRC ACP (C-Web Interface)
- Viewing General Statistics for SRC ACP (C-Web Interface) on page 308
- Viewing Device Statistics for SRC ACP (C-Web Interface) on page 309

Viewing Device Statistics for SRC ACP (C-Web Interface)

Purpose View device statistics for SRC ACP.

Action To view device statistics for SRC ACP:

1. Click **ACP>Statistics>Device**.

The Statistics/Device pane appears.

2. In the Filter box, enter a substring of the virtual router name, or leave the box blank to display information for all virtual routers.
3. In the Slot box, enter the number of the slot for which you want to display device statistics.
4. Select an output style from the Style list.
5. Click **OK**.

The Statistics/Device pane displays router statistics for SRC ACP.

Related Topics

- Configuring SRC ACP Properties (C-Web Interface)
- Viewing General Statistics for SRC ACP (C-Web Interface) on page 308
- Viewing Device Statistics for SRC ACP (C-Web Interface) on page 309

PART 6

Using External Subscriber Monitor

- [Configuring External Subscriber Monitor with the SRC CLI on page 313](#)
- [Monitoring External Subscriber Events with the SRC CLI on page 325](#)
- [Monitoring External Subscriber Events with the C-Web Interface on page 329](#)

CHAPTER 23

Configuring External Subscriber Monitor with the SRC CLI

This chapter describes how you can integrate IP address managers into an SRC-managed network so that the SAE is notified about subscriber events. Topics include:

- Overview of External Subscriber Monitor on page 313
- Configuring External Subscriber Monitor (SRC CLI) on page 314
- Configuring the NIC Proxy for the Pseudo-RADIUS Server (SRC CLI) on page 318
- Configuring the Pseudo-RADIUS Server for External Subscriber Monitor (SRC CLI) on page 320
- Configuring the Client Secret for External Subscriber Monitor (SRC CLI) on page 321
- Configuring Event Notification for External Subscriber Monitor (SRC CLI) on page 322
- Starting External Subscriber Monitor (SRC CLI) on page 323
- Stopping External Subscriber Monitor (SRC CLI) on page 323

Overview of External Subscriber Monitor

You use the External Subscriber Monitor application with the event notification method of logging in subscribers and creating subscriber sessions. You can use event notification when you integrate devices into the SRC network that do not notify the SAE about subscriber events, such as when a subscriber logs in or when the address assignment is terminated.

External Subscriber Monitor must view all RADIUS accounting messages associated with subscriber sessions. External Subscriber Monitor is stateless and cannot synchronize the current set of subscribers when there is a failure. If events are missed because of a software or network failure, the overall state recovers when RADIUS interim updates are sent. For example, missed ipUp events become effective when the next interim update is sent, and missed ipDown events time out after the configured RADIUS time to live.

External Subscriber Monitor is configured as a pseudo-RADIUS server and acts as a RADIUS accounting server. Configure the router or RADIUS server to duplicate accounting packets to External Subscriber Monitor. When External Subscriber Monitor is the pseudo-RADIUS server, it handles software failures more robustly. The pseudo-RADIUS

server does not acknowledge failed accounting requests and gives the RADIUS client the option to retransmit the accounting packet to a backup External Subscriber Monitor.

- Related Topics**
- For information about event notification with other third-party network devices, see Logging In Subscribers and Creating Sessions on page 93
 - Configuring External Subscriber Monitor (SRC CLI) on page 314
 - Starting External Subscriber Monitor (SRC CLI) on page 323
 - Configuring the Pseudo-RADIUS Server for External Subscriber Monitor (SRC CLI) on page 320
 - Configuring the Client Secret for External Subscriber Monitor (SRC CLI) on page 321
 - Configuring Event Notification for External Subscriber Monitor (SRC CLI) on page 322

Configuring External Subscriber Monitor (SRC CLI)

Configure initial properties, including directory connection and directory eventing properties.

Tasks to configure External Subscriber Monitor are:

1. Configuring Basic Local Properties for External Subscriber Monitor on page 314
2. Configuring Initial Properties for External Subscriber Monitor on page 315
3. Configuring Directory Connection Properties for External Subscriber Monitor on page 315
4. Configuring Eventing Properties for External Subscriber Monitor on page 316
5. Configuring Logging Destinations for External Subscriber Monitor on page 316

Configuring Basic Local Properties for External Subscriber Monitor

After you complete the configuration changes, restart External Subscriber Monitor for the configuration changes to take effect. Use the following configuration statements to configure basic local properties:

```
slot number external-subscriber-monitor {  
    java-garbage-collection-options java-garbage-collection-option;  
    java-heap-size java-heap-size;  
}
```

To configure basic local properties:

1. From configuration mode, access the configuration statement that configures the local properties.

```
user@host# edit slot 0 external-subscriber-monitor
```

2. Configure the garbage collection functionality of the Java Virtual Machine.

```
[edit slot 0 external-subscriber-monitor]  
user@host# set java-garbage-collection-options java-garbage-collection-options
```

3. (Optional) If you encounter problems caused by lack of memory, change the maximum memory size available to the JRE.

```
[edit slot 0 external-subscriber-monitor]
user@host# set java-heap-size java-heap-size
```

4. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor]
user@host# show
```

Configuring Initial Properties for External Subscriber Monitor

Use the following configuration statements to configure initial properties for External Subscriber Monitor:

```
slot number external-subscriber-monitor initial {
  dynamic-dn dynamic-dn;
}
```

To configure initial local properties:

1. From configuration mode, access the configuration statement that configures the initial properties.

```
user@host# edit slot 0 external-subscriber-monitor initial
```

2. Specify the properties for External Subscriber Monitor.

```
[edit slot 0 external-subscriber-monitor initial]
user@host# set ?
```

For more information about configuring local properties for the SRC components, see [Changing the Location of Data in the Directory](#).

Configuring Directory Connection Properties for External Subscriber Monitor

Use the following configuration statements to configure directory connection properties for External Subscriber Monitor:

```
slot number external-subscriber-monitor initial directory-connection {
  url url;
  backup-urls backup-urls...;
  principal principal;
  credentials credentials;
  timeout timeout;
  check-interval check-interval;
  blacklist;
  protocol (ldaps);
  snmp-agent;
}
```

To configure directory connection properties:

1. From configuration mode, access the configuration statement that configures the directory connection properties.

```
user@host# edit slot 0 external-subscriber-monitor initial directory-connection
```

2. Specify the properties for External Subscriber Monitor.

```
[edit slot 0 external-subscriber-monitor initial directory-connection]
```

```
user@host# set ?
```

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor initial directory-connection]
```

```
user@host# show
```

Configuring Eventing Properties for External Subscriber Monitor

Use the following configuration statements to configure directory eventing properties for External Subscriber Monitor:

```
slot number external-subscriber-monitor initial directory-eventing {  
    eventing;  
    signature-dn signature-dn;  
    polling-intervall polling-interval;  
    event-base-dn event-base-dn;  
    dispatcher-pool-size dispatcherr-pool-size;  
}
```

To configure directory eventing properties:

1. From configuration mode, access the configuration statement that configures the directory eventing properties.

```
user@host# edit slot 0 external-subscriber-monitor initial directory-eventing
```

2. Specify the initial directory eventing properties for External Subscriber Monitor.

```
[edit slot 0 external-subscriber-monitor initial directory-eventing]
```

```
user@host# set ?
```

For more information about configuring local properties for the SRC components, see [Configuring Initial Directory Eventing Properties for SRC Components](#).

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor initial directory-connection]
```

```
user@host# show
```

Configuring Logging Destinations for External Subscriber Monitor

Use the following configuration statements to configure directory logging destinations for External Subscriber Monitor:

```
slot number external-subscriber-monitor logger logger-name...  
slot number external-subscriber-monitor logger logger-name file {  
    filter filter;  
    filename filename;  
    rollover-filename rollover-filename;  
    maximum-file-size maximum-file-size;  
}  
slot number external-subscriber-monitor logger logger-name syslog {  
    filter filter;  
    host host;  
    facility facility;  
    format format;  
}
```

Configuring Logging Destinations to Store Messages in a File

To configure logging destinations to store log messages in a file:

1. From configuration mode, access the configuration statement that configures the name and type of logging properties. In this sample procedure, the logging destination called file-1 is configured.

```
user@host# edit slot 0 external-subscriber-monitor logger file-1 file
```

2. Specify the properties for the logging destination.

```
[edit slot 0 external-subscriber-monitor logger file-1 file]
user@host# set ?
```

For more information about configuring properties for the logging destination, see [Configuring a Component to Store Log Messages in a File \(SRC CLI\)](#).

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor logger file-1 file]
user@host# show
```

Configuring Logging Destinations to Send Messages to System Logging Facility

To configure logging destinations to send log messages to the system logging facility:

1. From configuration mode, access the configuration statement that configures the name and type of logging properties. In this sample procedure, the logging destination is called syslog-1.

```
user@host# edit slot 0 external-subscriber-monitor logger syslog-1 syslog
```

2. Specify the properties for the logging destination.

```
[edit slot 0 external-subscriber-monitor logger syslog-1 syslog]
user@host# set ?
```

For more information about configuring properties for the logging destination, see [Configuring System Logging \(SRC CLI\)](#).

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor logger file-1 file]
user@host# show
```

Related Topics

- [Configuring External Subscriber Monitor \(C-Web Interface\)](#)
- [Starting External Subscriber Monitor \(SRC CLI\) on page 323](#)
- [Viewing Statistics for External Subscriber Monitor \(C-Web Interface\) on page 329](#)
- [Overview of External Subscriber Monitor on page 313](#)

Configuring the NIC Proxy for the Pseudo-RADIUS Server (SRC CLI)

Configure the NIC proxy for the pseudo RADIUS server..

Tasks to configure the NIC proxy are:

1. Configuring Resolution Information for a NIC Proxy on page 318
2. Changing the Configuration for the NIC Proxy Cache on page 318
3. Configuring a NIC Proxy for NIC Replication on page 319

Configuring Resolution Information for a NIC Proxy

Use the following configuration statements to configure the NIC proxy:

```
slot number external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
  resolution {
    resolver-name resolver-name;
    constraints constraints;
  }
```

To configure resolution information for a NIC proxy:

1. From configuration mode, access the configuration statement that configures the NIC proxy configuration. In this sample procedure, the NIC proxy called radius-accounting-nic is configured.

```
user@host# edit slot 0 external-subscriber-monitor nic-proxy-configuration
radius-accounting-nic resolution
```

2. Specify the resolution information for this NIC proxy.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
resolution]
user@host# set ?
```

For more information about configuring resolution information for a NIC proxy, see “Configuring Resolution Information for a NIC Proxy (SRC CLI)” on page 166.

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
resolution]
user@host# show
```

Changing the Configuration for the NIC Proxy Cache

You can modify cache properties for the NIC proxy to optimize the resolution performance for your network configuration and system resources. Typically, you can use the default settings for the cache properties. The configuration statements are available at the Advanced editing level.

Use the following configuration statements to configure the NIC proxy cache:

```
slot number external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
  cache {
    cache-size cache-size;
```

```

cache-cleanup-interval cache-cleanup-interval;
cache-entry-age cache-entry-age;
}

```

To configure the cache for a NIC proxy:

1. From configuration mode, access the configuration statement that configures the NIC proxy configuration. In this sample procedure, the NIC proxy called radius-accounting-nic is configured.

```

user@host# edit slot 0 external-subscriber-monitor nic-proxy-configuration
radius-accounting-nic cache

```

2. Specify the cache properties for the NIC proxy.

```

[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
cache]
user@host# set ?

```

3. (Optional) Verify your configuration.

```

[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
cache]
user@host# show

```

Configuring a NIC Proxy for NIC Replication

Typically, you configure NIC replication to keep the NIC highly available. You configure NIC host selection to specify the groups of NIC hosts to be contacted to resolve a request, and to define how the NIC proxy handles NIC hosts that the proxy is unable to contact. The configuration statements are available at the Advanced editing level.

Use the following configuration statements to configure NIC host selection for a NIC proxy:

```

slot number external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection {
  groups groups;
  selection-criteria (roundRobin | randomPick | priorityList);
}
slot number external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection blacklisting {
  try-next-system-on-error;
  number-of-retries-before-blacklisting number-of-retries-before-blacklisting;
  blacklist-retry-interval blacklist-retry-interval;
}

```

To configure a NIC proxy to use NIC replication:

1. From configuration mode, access the configuration statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called radius-accounting-nic is configured.

```

user@host# edit slot 0 external-subscriber-monitor nic-proxy-configuration
radius-accounting-nic nic-host-selection

```

2. (Optional) Configure NIC host selection for a NIC proxy.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection]
user@host# set ?
```

For more information about configuring NIC host selection for a NIC proxy, see “Configuring a NIC Proxy for NIC Replication (SRC CLI)” on page 169.

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection]
user@host# show
```

4. Access the configuration statement that specifies the NIC proxy configuration for blacklisting—the process of handling nonresponsive NIC hosts.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection]
user@host# edit blacklisting
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection blacklisting]
```

5. (Optional) Configure blacklisting for a NIC proxy.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection blacklisting]
user@host# set ?
```

For more information about configuring NIC host selection for a NIC proxy, see “Configuring a NIC Proxy for NIC Replication (SRC CLI)” on page 169.

6. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection blacklisting]
user@host# show
```

- Related Topics**
- Configuring the NIC Proxy for the Pseudo-RADIUS Server (C-Web Interface)
 - Configuring the Pseudo-RADIUS Server for External Subscriber Monitor (SRC CLI) on page 320
 - Overview of External Subscriber Monitor on page 313

Configuring the Pseudo-RADIUS Server for External Subscriber Monitor (SRC CLI)

Use the following configuration statements to configure External Subscriber Monitor as a RADIUS accounting server:

```
slot number external-subscriber-monitor radius-accounting {
  port port;
  service-type (all | login | framed | callback-login | callback-framed | outbound |
    administrative | nas-prompt | authenticate-only | callback-nas-prompt | callback-check
    | callback-administrative);
  allow [allow...];
  deny [deny...];
  maximum-queue-length maximum-queue-length;
```



```
}
```

To configure the RADIUS accounting server:

1. From configuration mode, access the configuration statement that configures the RADIUS server.

```
user@host# edit slot 0 external-subscriber-monitor radius-accounting
```
2. (Optional) Specify the listening port for RADIUS requests.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# set port port
```
3. (Optional) Specify the service type of the RADIUS packets that will be forwarded.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# set service-type service-type
```
4. (Optional) Specify a list that filters which packets are forwarded to the SAE based on NAS ID or NAS IP.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# set allow [allow...]
```
5. (Optional) Specify a list that filters which packets are forwarded to the SAE based on NAS ID or NAS IP.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# set deny [deny...]
```
6. Specify the maximum number of unacknowledged RADIUS messages to be received from the RADIUS server before it discards new messages.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# set maximum-queue-length set maximum-queue-length
```
7. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# show
```

Related Topics

- Configuring the Pseudo-RADIUS Server for External Subscriber Monitor (C-Web Interface)
- Configuring External Subscriber Monitor (SRC CLI) on page 314
- Configuring Event Notification for External Subscriber Monitor (SRC CLI) on page 322
- Configuring the NIC Proxy for the Pseudo-RADIUS Server (SRC CLI) on page 318
- Viewing Statistics for External Subscriber Monitor (SRC CLI) on page 325

Configuring the Client Secret for External Subscriber Monitor (SRC CLI)

Use the following configuration statements to configure trusted clients for External Subscriber Monitor. If no clients are configured, all RADIUS accounting packets are discarded.

```
slot number external-subscriber-monitor radius-accounting client client-address {
```

```
secrets secret;  
}
```

To configure trusted clients for External Subscriber Monitor:

1. From configuration mode, access the configuration statement that configures the RADIUS server, and specify the client address.

```
user@host# edit slot 0 external-subscriber-monitor radius-accounting client  
client-address
```

2. Specify the shared secret of the RADIUS client.

```
[edit slot 0 external-subscriber-monitor radius-accounting]  
user@host# set secret secret
```

- Related Topics**
- Configuring the Client Secret for External Subscriber Monitor (C-Web Interface)
 - Configuring External Subscriber Monitor (SRC CLI) on page 314
 - Configuring the Pseudo-RADIUS Server for External Subscriber Monitor (SRC CLI) on page 320
 - Configuring Event Notification for External Subscriber Monitor (SRC CLI) on page 322
 - Overview of External Subscriber Monitor on page 313

Configuring Event Notification for External Subscriber Monitor (SRC CLI)

Use the following configuration statements to configure External Subscriber Monitor as a RADIUS accounting server:

```
slot number external-subscriber-monitor event-notification {  
  event-threads event-threads;  
  event-thread-idle-timeout event-thread-idle-timeout;  
  event-retry-timeout event-retry-timeout;  
  event-retry-interval event-retry-interval;  
  session-timeout session-timeout;  
}
```

To configure event notification

1. From configuration mode, access the configuration statement that configures the event notification.

```
user@host# edit slot 0 external-subscriber-monitor event-notification
```

2. (Optional) Specify the maximum number of concurrent threads in a pool for event handlers.

```
[edit slot 0 external-subscriber-monitor event-notification  
user@host# set event-threads event-threads
```

3. (Optional) Specify the time to keep an event handler alive for reuse.

```
[edit slot 0 external-subscriber-monitor event-notification  
user@host# set event-thread-idle timeout event-thread-idle-timeout
```

4. (Optional) Specify the maximum retry time before an event is discarded.

```
[edit slot 0 external-subscriber-monitor event-notification]
user@host# set event-retry-timeout event-retry-timeout.
```

5. (Optional) Specify the time to wait before the server retries failed events.

```
[edit slot 0 external-subscriber-monitor event-notification]
user@host# set event-retry-interval event-retry-interval
```

6. Specify the keepalive time for a RADIUS subscriber or service.

```
[edit slot 0 external-subscriber-monitor event-notification]
user@host# set session-timeout session-timeout
```

- Related Topics**
- Configuring Event Notification for External Subscriber Monitor (C-Web Interface)
 - Configuring External Subscriber Monitor (SRC CLI) on page 314
 - Configuring the Client Secret for External Subscriber Monitor (SRC CLI) on page 321
 - Overview of External Subscriber Monitor on page 313

Starting External Subscriber Monitor (SRC CLI)

To start External Subscriber Monitor:

- Start External Subscriber Monitor from its installation directory.

```
user@host# enable component extsubmon
```

- Related Topics**
- Starting External Subscriber Monitor (C-Web Interface)
 - Stopping External Subscriber Monitor (SRC CLI) on page 323
 - Configuring External Subscriber Monitor (SRC CLI) on page 314
 - Viewing Statistics for External Subscriber Monitor (SRC CLI) on page 325
 - Overview of External Subscriber Monitor on page 313

Stopping External Subscriber Monitor (SRC CLI)

To stop External Subscriber Monitor:

- Stop External Subscriber Monitor from its installation directory.

```
user@host# disable component extsubmon
```

- Related Topics**
- Stopping External Subscriber Monitor (C-Web Interface)
 - Starting External Subscriber Monitor (SRC CLI) on page 323
 - Viewing Statistics for External Subscriber Monitor (SRC CLI) on page 325
 - Overview of External Subscriber Monitor on page 313

CHAPTER 24

Monitoring External Subscriber Events with the SRC CLI

- Viewing Statistics for External Subscriber Monitor (SRC CLI) on page 325
- Monitoring Statistics for External Subscriber Monitor (SRC CLI) on page 326
- Viewing Statistics for External Subscriber Monitor Event Notifications (SRC CLI) on page 326
- Monitoring Statistics for External Subscriber Monitor Event Notifications (SRC CLI) on page 327
- Viewing Statistics for the Agent Process (SRC CLI) on page 328

Viewing Statistics for External Subscriber Monitor (SRC CLI)

Purpose View RADIUS accounting statistics for External Subscriber Monitor.

Action user@host> **show external-subscriber-monitor statistics radius-accounting**

Client Statistics

Client Address	10.227.7.45
Number of accounting start received	4
Number of accounting stop received	0
Number of accounting interim received	0
Number of discarded accounting requests	0

Meaning Table 14 on page 325 describes the output fields for the **show external-subscriber-monitor statistics radius-accounting** command. Output fields are listed in the order in which they appear.

Table 14: Output Fields for show external-subscriber-monitor statistics radius-accounting

Field Name	Field Description
Client Address	IP address of a RADIUS client. If not specified, displays statistics for all clients.
Number of accounting start received	Number of RADIUS start packets received.
Number of accounting stop received	Number of RADIUS stop packets received.
Number of accounting interim received	Number of RADIUS interim packets received.

Table 14: Output Fields for show external-subscriber-monitor statistics radius-accounting (*continued*)

Field Name	Field Description
Number of discarded accounting requests	Number of RADIUS packets discarded.

- Related Topics**
- Configuring External Subscriber Monitor (SRC CLI) on page 314
 - Viewing Statistics for External Subscriber Monitor (C-Web Interface) on page 329
 - Monitoring Statistics for External Subscriber Monitor (SRC CLI) on page 326
 - Viewing Statistics for External Subscriber Monitor Event Notifications (SRC CLI) on page 326
 - Viewing Statistics for the Agent Process (SRC CLI) on page 328

Monitoring Statistics for External Subscriber Monitor (SRC CLI)

Purpose Display real-time statistics for External Subscriber Monitor.

Action To display real-time statistics about RADIUS accounting for External Subscriber Monitor:

```
user@host> monitor external-subscriber-monitor radius-accounting client-address
client-address
```

To specify the time for refreshing the data:

```
user@host> monitor external-subscriber-monitor radius-accounting client-address
client-address interval interval
```

- Related Topics**
- Viewing Statistics for External Subscriber Monitor (SRC CLI) on page 325

Viewing Statistics for External Subscriber Monitor Event Notifications (SRC CLI)

Purpose View statistics for the External Subscriber Monitor event notifications.

Action user@host> show external-subscriber-monitor statistics event-notifications

```
Notification Statistics
Number of ipUp events      8
Number of ipDown events   0
Number of ipUp sent       0
Number of ipDown sent     0
Number of ipUp dropped    0
Number of ipDown dropped  4
Number of ipUp queued     0
Number of ipDown queued   0
Number of IpUp retries    0
Number of ipDown retries  0
```

Meaning Table 15 on page 327 describes the output fields for the **show external-subscriber-monitor statistics event-notifications** command. Output fields are listed in the order in which they appear.

Table 15: Output Fields for show external-subscriber-monitor statistics event-notifications

Field Name	Field Description
Number of ipUp events	Total number of ipUp notification events received, including ipUp sent, ipUp dropped, and ipUp queued
Number of ipDown events	Total number of ipDown notification events received, including ipDown sent, ipDown dropped, and ipDown queued
Number of ipUp sent	Total number of ipUp notification events successfully sent
Number of ipDown sent	Total number of ipDown notification events successfully sent
Number of ipUp dropped	Total number of ipUp notification events dropped due to network failure or difficulties locating managed SAE
Number of ipDown dropped	Total number of ipDown notification events dropped due to network failure or difficulties locating managed SAE
Number of ipUp queued	Total number of ipUp notification events queued to send to SAE
Number of ipDown queued	Total number of ipDown notification events queued to send to SAE
Number of IpUp retries	Total number of ipUp notification events resent tries
Number of IpDown retries	Total number of ipDown notification events resent tries
Number of Nic lookup retries	Total number of NIC lookup retries

- Related Topics**
- Configuring Event Notification for External Subscriber Monitor (SRC CLI) on page 322
 - Viewing Statistics for External Subscriber Monitor Event Notifications (C-Web Interface) on page 330
 - Monitoring Statistics for External Subscriber Monitor Event Notifications (SRC CLI) on page 327
 - Viewing Statistics for External Subscriber Monitor (C-Web Interface) on page 329
 - Viewing Statistics for the Agent Process (SRC CLI) on page 328

Monitoring Statistics for External Subscriber Monitor Event Notifications (SRC CLI)

Purpose Display real-time statistics about event notifications for External Subscriber Monitor.

Action To display real-time statistics about event notifications for External Subscriber Monitor:

```
user@host> monitor external-subscriber-monitor event-notifications
```

To specify the time for refreshing the data:

```
user@host> monitor external-subscriber-monitor event-notifications interval
interval
```

- Related Topics**
- Viewing Statistics for External Subscriber Monitor Event Notifications (SRC CLI) on page 326

Viewing Statistics for the Agent Process (SRC CLI)

Purpose View statistics for the agent process.

Action user@host> **show external-subscriber-monitor statistics process**

Process Statistics

```
Up Time      Time1147 seconds since Thu Jan 31 15:56:39 EST 2008
Threads      246
Heap In Use   use142343 kilo bytes
Heap Limit    1012672 kilo bytes
```

Meaning Table 16 on page 328 describes the output fields for the **show external-subscriber-monitor statistics process** command. Output fields are listed in the order in which they appear.

Table 16: Output Fields for show external-subscriber-monitor statistics process

Field Name	Field Description
Up time	Length of time the agent has been running on the system. Includes the date and time at which the agent was last started.
Threads	Number of threads in use.
Heap In Use	Heap size allocated by the Java Virtual Machine. The percentage indicates the percentage of the heap in use. We recommend that if the percent in use is more than 90% additional heap be allocated.
Heap Limit	Size of Java heap configured.

- Related Topics**
- Viewing Statistics for External Subscriber Monitor (C-Web Interface) on page 329
 - Viewing Statistics for External Subscriber Monitor Event Notifications (SRC CLI) on page 326
 - Viewing Statistics for the Agent Process (C-Web Interface) on page 330

CHAPTER 25

Monitoring External Subscriber Events with the C-Web Interface

- Viewing Statistics for External Subscriber Monitor (C-Web Interface) on page 329
- Viewing Statistics for External Subscriber Monitor Event Notifications (C-Web Interface) on page 330
- Viewing Statistics for the Agent Process (C-Web Interface) on page 330

Viewing Statistics for External Subscriber Monitor (C-Web Interface)

Purpose View statistics for External Subscriber Monitor.

- Action**
1. Click **Monitor>Ext Sub Monitor>Statistics>RADIUS Accounting**.
The Statistics/RADIUS Accounting pane appears.
 2. In the Client Address box, enter the address of the client for which you want to view statistics.
 3. Select an output style from the Style list.
 4. Click **OK**.
The Statistics/RADIUS Accounting pane displays the RADIUS statistics for External Subscriber Monitor.

- Related Topics**
- Configuring External Subscriber Monitor (C-Web Interface)
 - Viewing Statistics for External Subscriber Monitor (SRC CLI) on page 325
 - Viewing Statistics for External Subscriber Monitor Event Notifications (C-Web Interface) on page 330

- Viewing Statistics for the Agent Process (SRC CLI) on page 328

Viewing Statistics for External Subscriber Monitor Event Notifications (C-Web Interface)

Purpose View statistics for the External Subscriber Monitor notifications.

Action 1. Click **Monitor>Ext Sub Monitor>Statistics>Event Notification**.

The Statistics/Event Notification pane displays the event notification statistics for the External Subscriber Monitor.

Related Topics

- Configuring Event Notification for External Subscriber Monitor (SRC CLI) on page 322
- Viewing Statistics for External Subscriber Monitor Event Notifications (C-Web Interface) on page 330
- Viewing Statistics for External Subscriber Monitor (SRC CLI) on page 325

Viewing Statistics for the Agent Process (C-Web Interface)

Purpose View statistics for the agent process.

Action 1. Click **Monitor>Ext Sub Monitor>Statistics>Process**.

The Statistics/Process pane displays the process statistics for the agent.

Related Topics

- Viewing Statistics for the Agent Process (SRC CLI) on page 328
- Viewing Statistics for External Subscriber Monitor (SRC CLI) on page 325
- Viewing Statistics for External Subscriber Monitor Event Notifications (C-Web Interface) on page 330

PART 7

Using Session State Registrar

- Session State Registrar Overview on page 333
- Planning Your Session State Registrar Cluster on page 355
- Configuring the Session State Registrar (SRC CLI) on page 363
- Managing the SSR Cluster on page 381
- Monitoring the SSR Cluster on page 385

Session State Registrar Overview

- Overview of the Session State Registrar on page 333
- SSR Node Types on page 334
- SSR Node Groups on page 335
- C Series Controller Requirements on page 336
- SSR Cluster Configurations Overview on page 337
- Scaling the SSR Cluster on page 338
- SSR Cluster Network Requirements on page 339
- Supported SSR Cluster Configurations on page 341
- SSR Database Schema on page 347
- Overview of Making Modifications to the SSR Database Schema on page 351
- SSR Database Operating Modes on page 351
- Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 352

Overview of the Session State Registrar

The Session State Registrar (SSR) solution implements a stateless, highly reliable and highly available cluster. It separates front end processes from back-end data functions that take place on two or four data servers. Multiple C Series Controllers collaborate and perform different aspects of operation within the cluster to provide a common sessions database in a highly-available, redundant environment. The common shared resources of the cluster can be accessed simultaneously by up to twenty-four C Series Controllers acting as SSR clients.

The front-end SSR client hosts and SSR back-end data servers collaborate to provide:

- High availability
- Session state preservation during failover of front-end client nodes
- Application session awareness

When used in conjunction with an MX Series router running the packet-triggered subscribers and policy control (PTSP) solution, the SSR stores the IP edge attachment sessions learned from IP edge devices in the centralized SSR database. The IP edge

session stored in the SSR database can be used by the SAE to map the sessions received from the MX Series router. An IP edge session is uniquely identified by IP address and VPN ID, and includes subscriber identity information, which is used to locate the subscriber profile for MX sessions that have the same subscriber IP address.

To work efficiently and to provide redundancy, a production SSR cluster must be built on a fast, isolated, redundant network infrastructure. It must include multiple hosts so a single point of failure does not prevent the cluster from operating.

- Related Topics**
- [SSR Node Types on page 334](#)
 - [SSR Node Groups on page 335](#)
 - [C Series Controller Requirements on page 336](#)
 - [SSR Cluster Configurations Overview on page 337](#)

SSR Node Types

An SSR cluster has both a physical and logical organization. A C Series Controller that is a member of the cluster is called a *node*. A process that runs on a node and is part of the cluster is called a *component*. These terms are not interchangeable.

Two types of nodes are included in an SSR cluster, each with a specific role within the cluster:

- A *data node* is a C Series Controller residing in the back end of the cluster. It runs the data storage engine component, which cooperatively manages, replicates, and stores data in the SSR storage engine with other data nodes. Each data node has its own memory and permanent storage, and maintains both a portion of the working copy of the SSR database and a portion of one or more replicas of the SSR database. A cluster can contain either two or four data nodes.
- A *client node* is a C Series Controller that resides in the front end of the SSR cluster. Client nodes are responsible for several functions, including hosting SSR client components, hosting the SSR database front-end component, and optionally hosting the *management server* component.

The client components are the SRC components such as the Subscriber Information Collector (SIC), the Network Information Collector (NIC), the Service Activation Engine (SAE), the IMS Services Gateway, Dynamic Service Activator and other SRC components.

In addition to hosting the SSR client components, each client node hosts a front-end database component that reads and writes data into the SSR database and manipulates the cluster's shared data which is hosted by the data nodes.

At least one client node must host a *management server*. The management server controls itself and all data nodes in the cluster. So that there is no single point of failure, you should configure a minimum of two client nodes to host management servers.

To uniquely identify the various components running in the cluster, the SRC software assigns a node ID number to each component in the node. Node IDs are generated automatically by the SRC software and cannot be modified.

- Related Topics**
- Overview of the Session State Registrar on page 333
 - SSR Node Groups on page 335
 - C Series Controller Requirements on page 336
 - Configuring the Nodes in the SSR Cluster (SRC CLI) on page 368

SSR Node Groups

Each data node participates in a *node group* of two data nodes. A cluster with two data nodes has a single node group; a cluster with four data nodes has two node groups, each with two data nodes. Each node group stores different partitions and replicas.

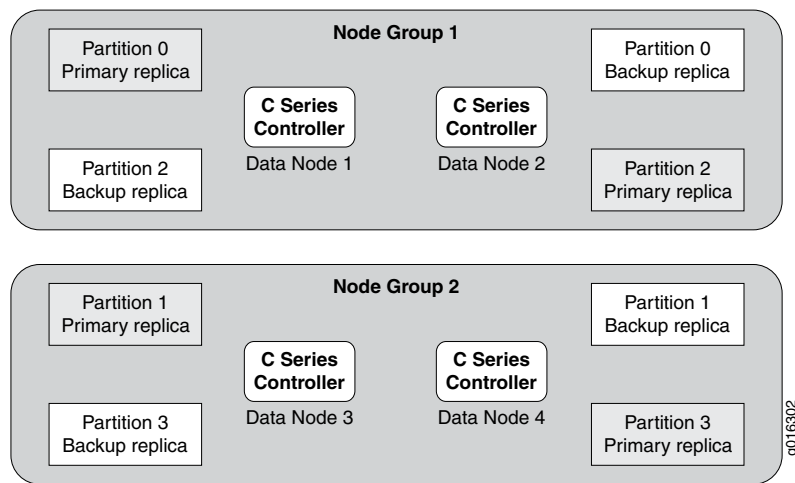
- A *partition* is a portion of all the data stored by the cluster. There are as many cluster partitions as node groups in the cluster. Each node group keeps at least one copy of any partitions assigned to it (that is, at least one replica) available to the cluster.
- A *replica* is a copy of a partition. Each data node in a node group stores a replica of a partition. A replica belongs entirely to a single data node; a node can (and usually does) store several replicas because maintaining two replicas is the fixed setting for the SSR.

Figure 53 on page 336 shows the data elements of an SSR cluster with four data nodes arranged in two node groups of two nodes each. Nodes 1 and 2 belong to Node Group 1. Nodes 3 and 4 belong to Node Group 2.

- Because there are four data nodes, there are four partitions.
- The number of replicas is two, to create two copies of each primary partition.

So long as both nodes in one node group are operating, or one node in each node group is operating, the cluster remains viable.

Figure 53: SSR with Four Data Nodes in Two Groups



The data stored by the cluster in Figure 53 on page 336 is divided into four partitions: 0, 1, 2, and 3. Multiple copies of each partition are stored within the same node group. Partitions are stored on alternate node groups:

- Partition 0 is stored on Node Group 1. A primary replica is stored on Data Node 1 and a backup replica is stored on Data Node 2.
- Partition 1 is stored on the other node group, Node Group 2. The primary replica is on Data Node 3 and its backup replica is on Data Node 4.
- Partition 2 is stored on Node Group 1. The placement of its two replicas is reversed from that of Partition 0; the primary replica is stored on Data Node 2 and the backup on Data Node 1.
- Partition 3 is stored on Node Group 2, and the placement of its two replicas are reversed from those of partition 1: the primary replica is on Data Node 4 and the backup on Data Node 3.

- Related Topics**
- C Series Controller Requirements on page 336
 - SSR Cluster Configurations Overview on page 337
 - Supported SSR Cluster Configurations on page 341
 - Scaling the SSR Cluster on page 338

C Series Controller Requirements

An SSR cluster does not have any single point of failure; each node (C Series Controller) in the cluster has its own memory and disks.

All nodes in the cluster require at least two physical Ethernet ports that provide the same throughput. Bonding the Ethernet interfaces to a single IP address is required. You can accomplish this using SRC group interfaces. Data nodes require 1000Base-T (gigabit Ethernet). For client nodes 100Base-T is sufficient.



NOTE: All data nodes must have equal processor power, memory space, and available bandwidth because they are tightly coupled and share data. If the overall throughput of the data nodes varies from node to node, performance degrades. Therefore, all data nodes must be of the same C Series Controller model, either all C2000 model or all C4000 model.

- Related Topics**
- SSR Node Types on page 334
 - SSR Cluster Configurations Overview on page 337
 - Supported SSR Cluster Configurations on page 341
 - Planning the SSR Cluster Topology on page 355
 - SSR Cluster Network Requirements on page 339

SSR Cluster Configurations Overview

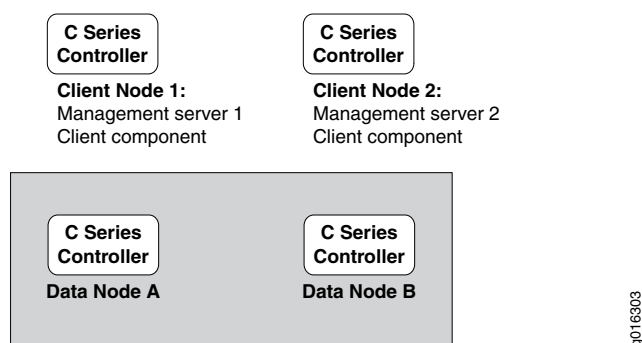
For the highest level of redundancy, each data node must run on its own C Series Controller. Each C Series Controller configured as a client node supports the SSR client components and at least one client node must be configured to host the management server. The management server can run only on a client node. Separation is required so that management arbitration services continue if one of the data node fails. However, for full redundancy, two client nodes should be configured to host management servers.

Using these separation guidelines, we recommend a minimum SSR cluster size of:

- Two client nodes, each running an instance of the management server and a client component
- Two data nodes, each running on its own C Series Controller

This configuration is shown in Figure 54 on page 337:

Figure 54: Basic Session State Registrar Cluster



- Related Topics**
- C Series Controller Requirements on page 336
 - Scaling the SSR Cluster on page 338

- [SSR Cluster Network Requirements on page 339](#)
- [Planning the SSR Cluster Topology on page 355](#)

Scaling the SSR Cluster

You scale your SSR cluster by adding separately licensed Expansion Kits to the Starter Kit. The Starter Kit licenses you for the minimum cluster configuration of two client nodes, each hosting a management server and two data nodes. Expansion Kits are available to scale both the back end and front end of the cluster.

Scaling the Front End of the Cluster

You scale the front end of your SSR cluster by adding licenses for additional client nodes. Optionally, you can add a Management Server Expansion Kit, which allows you to add a third management server component to the on a client node. Each management server component must run on a separate client node.

The service capacity of the SSR cluster grows when you add additional client nodes to the front end. Adding additional client nodes, each of which can host an SSR client component, increases the resiliency of the cluster and the speed of processing a particular transaction because wait time is reduced. Up to twenty four client nodes are supported. At least one of the client nodes must be configured to host the management server component. For redundancy, at least two client nodes must be configured to host the management server component.

The client nodes do not require identical configurations; they can be configured with different components or communications interfaces. For example, one client node might host the Subscriber Information Collector (SIC) component used for the PTSP feature. Another client node might host a different SRC component, such as the SAE or NIC. However, to ensure no single point of failure, we recommend that you configure your cluster with enough client nodes to provide redundancy of the components. For example, for redundancy in a cluster running the SIC component, you would want to have at least:

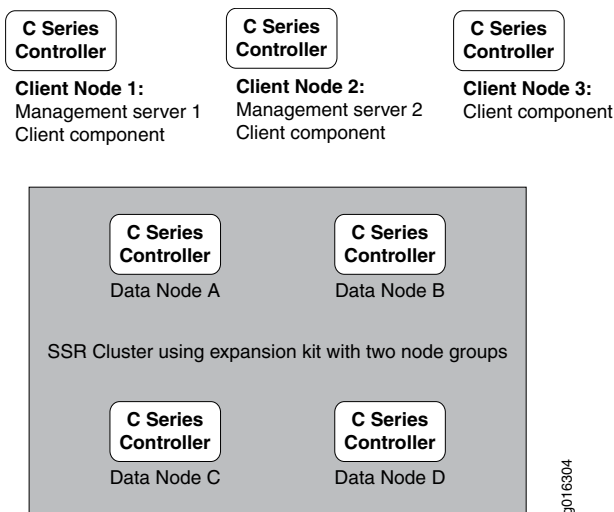
- Two client nodes, each hosting the SIC component and the management server component
- Two data nodes

Scaling the Back End of the Cluster

You scale the back end of the cluster by adding a data node Expansion Kit which licenses you for two additional data nodes, bringing the total number of data nodes to four, which is the maximum allowed. The additional data nodes form a second node group in the example shown in Figure 55 on page 339, that provides more working memory for the SSR shared database. Each node group manages a partition of the primary SSR database and replicas. The data in each partition is synchronously replicated between the group's data nodes, so if one data node fails, the remaining node can still access all the data. This configuration also provides very quick failover times if a node fails.

Node groupings are managed by the management server. Node groups may not be formed in the same way shown in Figure 55 on page 339. For example, it is possible a new node and an existing node could form one group and the other nodes form another group.

Figure 55: SSR Cluster with Four Data Nodes Forming Two-Node Groups



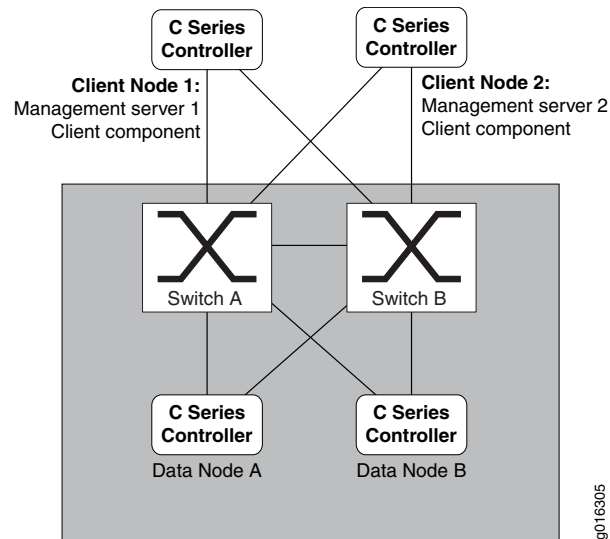
- Related Topics**
- [SSR Cluster Configurations Overview on page 337](#)
 - [SSR Cluster Network Requirements on page 339](#)
 - [Supported SSR Cluster Configurations on page 341](#)
 - [Planning the SSR Cluster Topology on page 355](#)
 - [SSR Cluster Planning Worksheets on page 357](#)

SSR Cluster Network Requirements

A redundant cluster requires a redundant network. We require dual interface cards in each C Series Controller. Use the SRC group interfaces feature to bond the two interfaces to a single IP address.

We recommend that the network be a dedicated subnet with dual switches. This fully duplicates the network, and each C Series Controller in the cluster has at least two routes to all other C Series Controller, as shown in Figure 56 on page 340.

Figure 56: SSR Cluster with Redundant Network



The SSR database schema uses primary key lookups as often as possible during transaction processing, so the database cluster performance scales almost linearly based on the number of data nodes in the cluster.

Do not configure the subnet to be shared beyond the cluster C Series Controllers, because communications between nodes are not encrypted or shielded in any way. The only means of protecting transmissions within a cluster is to run your the cluster on a protected network; do not interpose firewalls between any of the nodes.

Running the cluster on a private or protected network also increases efficiency because the cluster has exclusive use of all bandwidth between cluster nodes. This protects the cluster nodes from interference caused by transmissions between other devices on the network.

The SSR cluster requires Gigabit Ethernet between data nodes and the switch. Client nodes to the switch can use 100Base-T but Gigabit Ethernet is recommended. Network latency can severely degrade performance, as shown in Table 17 on page 340, so we also recommend that all servers be close enough together that latency is always much less than 10 ms.

Table 17: Latency Between Servers and Its Effect on Performance

Latency Times	Performance Degradation
0 ms latency (LAN)	Baseline performance as designed
10 ms latency	Up to 40% performance loss
20 ms latency	Up to 60% performance loss
More than 20 ms latency	Not supported

- Related Topics**
- C Series Controller Requirements on page 336
 - SSR Cluster Configurations Overview on page 337
 - Scaling the SSR Cluster on page 338
 - Supported SSR Cluster Configurations on page 341
 - Planning the SSR Cluster Topology on page 355

Supported SSR Cluster Configurations

The minimum configuration for a redundant SSR cluster is two client nodes, one of which must host a management server, and two data nodes. A maximum of twenty-four client nodes is supported. For redundancy, at least two client nodes should be configured to host management server components. Data nodes must be added in pairs. The maximum number of data nodes in a cluster is four.

Table 18 on page 341 lists the possible configurations.



CAUTION: Setting up an unsupported configuration can put data and equipment at risk and is not supported by Juniper Networks.

Also, note the latency limitation in Table 17 on page 340. We do not support cluster configurations with latency between nodes that exceeds 20 ms, as can occur if servers are set up to spread a cluster across widely separated locations.

Table 18: Supported Cluster Configurations

Data Nodes	Client Nodes
<p>Two</p> <ul style="list-style-type: none"> • Each running on its own C Series Controller 	<p>Up to 24</p> <ul style="list-style-type: none"> • Each running on its own C Series Controller • Up to three configured to run management servers • Each hosting SSR client components such as the SIC, SAE and so on • Minimum configuration is one client node/management server with one SSR client component (no redundancy)
<p>Four</p> <ul style="list-style-type: none"> • Each running on its own C Series Controller 	<p>Up to 24</p> <ul style="list-style-type: none"> • Each running on its own C Series Controller • Up to three configured to run management servers • Each hosting SSR client components such as the SIC, SAE and so on • Minimum configuration is one client node/management server with one SSR client component (no redundancy)

Failover Overview

To continue functioning without a service interruption after a component failure, a cluster requires at least 50 percent of its data nodes and client nodes running the management server component to be functional. If more than 50 percent of the data nodes fail, expect a service interruption, but continued operation of the available nodes.

Because SSR client components function as front ends to the back-end data storage portion of the cluster, they are not involved in any failover operations performed by the back-end data components. However, as an administrator, you need to ensure that the front end environment is configured so that it can survive the loss of components.

A data cluster prepares for failover automatically when the cluster starts. During startup, two events occur:

- One of the data nodes (usually the node with the lowest node ID) becomes the *master* of the node group. The master node stores the authoritative copy of the database.
- One data node or management server is elected *arbiter*. The arbiter is responsible for conducting elections among the surviving nodes to determine roles in the event of node failures.

In a cluster, each management server and data node are allocated a vote that is used during this startup election and during failover operations. One management server is selected as the initial arbiter of failover problems and of elections that result from them.

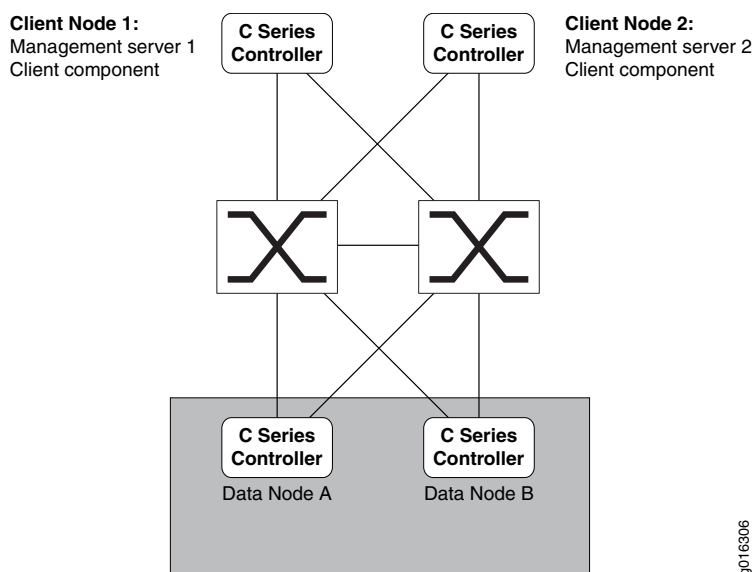
Within the cluster, data nodes and any client nodes hosting management servers monitor each other to detect communications loss. When either type of failure is detected, as long as nodes with more than 50 percent of the votes are operating, there is instantaneous failover and no service interruption. If exactly 50 percent of nodes and votes are lost, and if a data node is one of the lost nodes, the cluster determines which half of the database is to remain in operation. The half with the arbiter (which usually includes the master node) stays up, and the other half shuts down to prevent each node or node group from updating information independently.

When a failed data node (or nodes) returns to service, the working nodes resynchronize the current data with the restored nodes so all data nodes are up to date. How quickly this takes place depends on the current load on the cluster, the length of time the nodes were offline, and other factors.

Failover Examples

The following examples are based on the deployment of two client nodes each hosting a management server, and two data nodes set up with the recommended redundant network as shown in Figure 57 on page 343. Each client node is running a client component. The cluster is set up in a single data center on a fully switched, redundant, layer 2 network. Each of the nodes is connected to two switches using Ethernet bonding for interface failover. The switches have a back-to-back connection.

Figure 57: SSR Cluster with Redundant Network



g016306

Possible Failure Scenarios

With this basic configuration, a high level of redundancy is supported. So long as one data node is available to one client node/management server, the cluster is viable and functional.

- If either client node/management server (1 or 2), goes down, the effect on the facility and cluster is:
 - No impact to the SSR client component.
 - Devices (depending on the failover mechanism in the device) switch to their secondary targets—the remaining SSR client node. Recovery of the device when the failed client node returns to service depends on device implementation.
- If either data node A or B goes down, the effect is:
 - No service impact to the SSR client component component; both client nodes continue operation using the surviving data node.
 - The management servers running on the client nodes and the surviving data node detect that one data node has gone down, but no action is required, because failover is automatic.
 - When the failed data node returns to service, it synchronizes its data with the surviving data node and resumes operation.
- If both management servers running on client nodes 1 and 2 go down, the effect is:
 - No service impact to the SSR client components, because all client components and data nodes are still available. The data nodes continue to update themselves.
- If both data nodes go down, the effect on:

- The management servers is minimal. They detect that the data nodes are offline, but can only monitor them.
- The SSR client components running on the client nodes varies:
 - Sessions that do not require shared resources continue uninterrupted.
 - Sessions that require shared resources are rejected.

The client nodes continue to operate this way until the back-end data cluster comes back online; the cluster resumes normal operation using the data cluster automatically.

- If one half of the cluster (client node 1 and management server 1, and data node A or client node 2 and management server 2, and data node B) go down, the effect is:
 - No service impact, because a client node/management server and a data node are all still in service. Devices using the SSR client component in the failed client node fail over to the other SSR client component in the surviving client node.
 - When the failed data node returns to service, it synchronizes and updates its data with the surviving data node and resumes operation.
 - When the failed client node/management server returns to service, the devices assigned to use it as a primary resource return to service depending on the device implementation.

Distributed Cluster Failure and Recovery

You can divide a cluster and separate two equal halves between two data centers. In this case, the interconnection is made by dedicated communications links (shown as red lines in Figure 58 on page 345 and Figure 59 on page 346) that may be either:

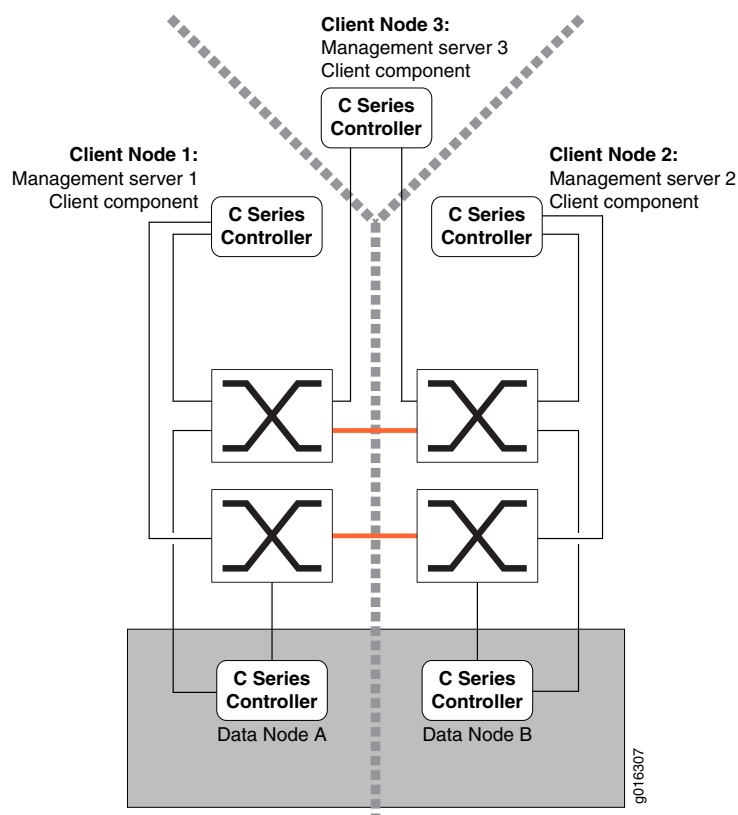
- A switched layer 2 network, just as the single site cluster is set up.
- A routed layer 3 network that uses a routing table with backup routes to route over multiple links between data centers.

However, separating the cluster like this creates a configuration that is vulnerable to a catastrophic failure that severs the two halves of a dispersed cluster. We recommend adding a third client node/management server at a location that has a separate alternative communication route to each half. A third client node/management server:

- Eliminates the possibility of the cluster being evenly split by a communications failure.
- Creates an odd number of votes for elections, which greatly reduces the need for arbitration.

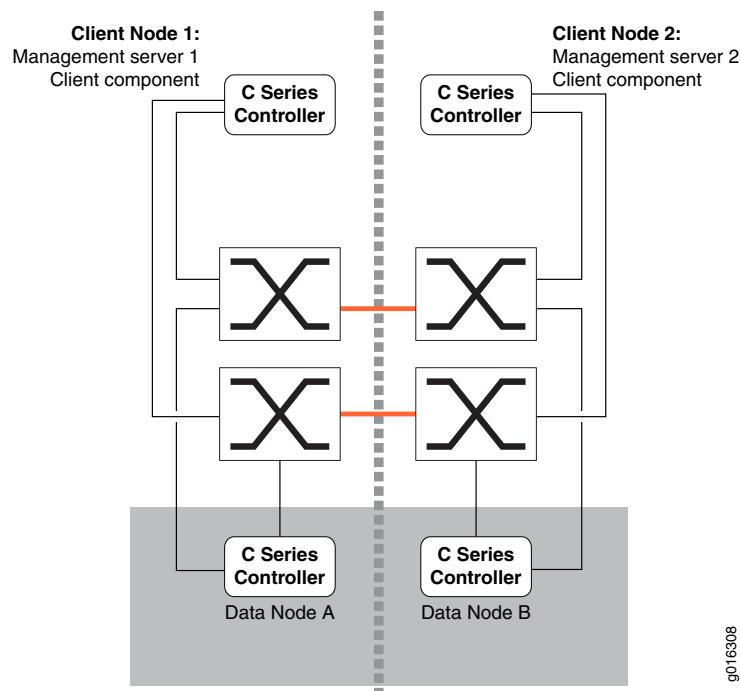
With a third client node/management server in place, failover in the dispersed cluster is well managed because one side of the cluster does not have to determine what role to assume. Recovery is likely to be quicker when the data nodes are reunited because each node's status is more likely to have been monitored by at least one management server that is in communication with each segment.

Figure 58: SSR Cluster Divided Between Two Sites with Tertiary Management Server



Without a third client node/management server, the configuration shown in Figure 59 on page 346 is vulnerable to data loss if both communication links are severed or if the nodes in the master half of the cluster all go offline simultaneously.

Figure 59: SSR Cluster Evenly Divided Between Two Sites



If either of those calamities occur, exactly half the nodes in a side survive. If the master nodes are operating on one or both sides, the cluster continues to function. But the secondary side cannot determine whether the master side is really no longer available, because it only has two votes. It can take 10-15 minutes for the secondary side of the cluster to automatically restart, promote itself to master status, and resume cluster operations.

The SSR client nodes connecting to the secondary side continue to work. However, modifications made to the SSR database may create a divergence between the two copies of the database. The longer the cluster is split, the greater the divergence, and the longer it takes to resolve when recovery takes place.

To eliminate these problems, we recommend a proven alternative: adding another client node with a third management server in a third location that can communicate with each half of the dispersed cluster. Without the tertiary client node/management server, there is a possibility of down time in a dispersed cluster that suffers a catastrophic failure.

If you cannot add a third client node/management server, we recommend that you configure the secondary side of the cluster not to automatically restart, but to go out of service when it instantaneously disconnects from the master side nodes. Then you can determine the best course of action—to keep the cluster offline or to promote the secondary side of the cluster, relink the client node/management servers, and be aware that reconciling the divergence must be part of the recovery procedure.

When the cluster is reunited and goes into recovery mode, the master and secondary data nodes attempt to reconcile the divergence that occurred during separation. The moment they come in contact, transitory failures appear on the client nodes because

the cluster configuration has changed; any transactions that are pending at that moment are aborted. The client nodes retry those transactions because they are classified as temporary failures; in most situations they are accepted on the first retry.

- Related Topics**
- Scaling the SSR Cluster on page 338
 - SSR Cluster Network Requirements on page 339
 - C Series Controller Requirements on page 336
 - Planning the SSR Cluster Topology on page 355
 - SSR Cluster Planning Worksheets on page 357

SSR Database Schema

When used in conjunction with an MX Series Router running the PTSP feature, the SSR stores the IP edge attachment sessions learned from IP edge devices in the SSR centralized database. The IP edge session stored in the SSR database can be used by the SAE to map the sessions received from the MX router service node. An IP edge session is uniquely identified by IP address and VPN ID. It includes subscriber identity information, which is used to locate the subscriber profile for MX sessions that have the same subscriber IP address.

The SSR cluster uses a relational SQL design to store data. The database schema includes multiple tables for storing subscriber identity information for the session and service activation requests.



NOTE: Changes to the SSR database schema requires you to restart all SRC components.

Subscriber Sessions Table

The format of subscriber identity information stored for each session is managed by the subscriber sessions table, which is controlled by the cluster's client node/management servers; the configuration of the subscriber sessions table is copied to all nodes in a node group, so all nodes operate with the same information. The default format of the subscriber sessions table, described in Table 19 on page 347, addresses the needs of most carriers; however, you can modify certain fields to address unique needs and situations.

Table 19: Subscriber Sessions Table Default Fields

Field Name	Field Type	Default Value	Notes
UserIPAddress	binary(4)	NOT NULL	Primary key. This column is fixed and is only resizable

Table 19: Subscriber Sessions Table Default Fields (*continued*)

Field Name	Field Type	Default Value	Notes
VpnID	varchar(16) CHARACTER SET utf8	NOT NULL, DEFAULT ""	Primary key. This column is fixed and is only resizable
UserName	varchar(24) CHARACTER SET utf8	DEFAULT NULL	Indexed by default
IMSI	varchar(15)	DEFAULT NULL	For storing International Mobile Subscriber Identity (IMSI)
CallingStationID	varchar(24) CHARACTER SET utf8	DEFAULT NULL	For storing Mobile Station International Subscriber Directory Number (MSISDN) or Calling-Station-ID
CalledStationID	varchar(60) CHARACTER SET utf8	DEFAULT NULL	For storing Access Point Name (APN) or Called-Station-ID
DeviceType	varchar(10) CHARACTER SET utf8	DEFAULT NULL	For storing International Mobile Equipment Identity (IMEI)
AccessType	varchar(16) CHARACTER SET utf8	DEFAULT NULL	For storing radio access technology (RAT)
SessionStartTime	Timestamp	NOT NULL	Start time of subscriber session This column is fixed and is only resizable
State	tinyint unsigned	NOT NULL	State of the subscriber session (1 for started, 2 for stopped) This column is fixed and is only resizable

The primary keys of the subscriber sessions table are the UserIpAddress and VpnID fields. The UserIpAddress field stores the subscriber's IP address in binary format. The default schema uses 4 bytes, which is sufficient for IPv4 addresses. You can modify the length of the UserIpAddress field to 16 if you are using IPv6 addresses. The VpnID field stores the address realm where the user IP address is unique. For non-VPN sessions, the VpnID must be set to its default value, which is an empty string.

The subscriber sessions table is configurable with some restrictions. You can add new columns, remove existing columns, or modify column length, type or index. You cannot

remove the UserIpAddress, VpnID, SessionStartTime, or SessionState columns; however, you can modify the length of the UserIpAddress and VpnID fields.

Attribute Associations

SSR client components such as the SAE, NIC, DSA and SIC, need to read and write information to the subscriber sessions table. To support this, you need to specify how SSR client component-specific attributes are translated to subscriber sessions table attributes by defining the mapping between the attributes.

This mapping provides a virtual schema composed of SAE plugin attributes. The virtual schema is available to all SSR client components, for example the SAE, NIC, DSA, SIC, and so on. These components use SAE plugin attributes in the virtual schema to access the subscriber sessions table. The attribute association mapping provides the correlation between the SAE plugin attributes and the attributes (columns) in the subscriber sessions table.

Use the **shared database cluster primary attribute-associations entity** configuration statement to define the mapping.

Following are several mapping examples.

```
shared database cluster primary attribute-associations {
  table subscriber-sessions field vpn-id{
    sae-plug-in-attribute vpn-id;
  }
}
```

For multivalued dictionary-type SAE plugin attributes, including PA_PROPERTY and PA_SUBSTITUTION, the suffix can be used to map a specific property or substitution to a field in the SSR subscriber sessions table. For example:

```
shared database cluster primary attribute-associations {
  table subscriber-sessions field called-station-id{
    sae-plug-in-attribute property.calledStationId;
  }
  table subscriber-sessions field calling-station-id{
    sae-plug-in-attribute property.callingStationId;
  }
}
```

In this example, the PA_PROPERTY plugin attribute contains two properties, calledStationId and callingStationId. The calledStationId property is mapped to the called-station-id field in the subscriber sessions table, and the callingStationId is mapped to the calling-station-id field. It is also possible to map a multivalued SAE plugin attribute to a field in the subscriber sessions table. Multiple values are concatenated together using a separator, and stored in the SSR field. When read from the SSR database, the multivalued attribute is restored from the field.

Service Sessions Table

The service sessions table stores service activation requests received through the Application Services Gateway (ASG) used by the MX Series Router PTSP feature to create a service session in the SSR. When the ASG receives a service activation request,

it queries the subscriber sessions table using the subscriber's VpnID and IP address, which are either received from the request directly or obtained through network information collector (NIC) lookup if the subscriber ID in the request is not VpnID+IP address. If the specified VpnID+IP address does not exist, the ASG replies with an unknown subscriber error. Otherwise, the ASG stores the service activation request with the attributes VpnID, UserIPAddress and SessionStartTime, which is taken from the attachment session record, along with SubscriptionName, SessionName, and activation attributes from the request. The SAE is notified through the SSR event interface when the service sessions table is updated: a record is created, updated or removed. The SAE also queries the service sessions table for service activations when it starts a new PTSP session. Table 20 on page 350 describes the fields and default values for the service sessions table.

Table 20: Service Sessions Table Default Fields

Field Name	Field Type	Default Value	Notes
UserIPAddress	binary(4)	NOT NULL	Primary key
VpnID	varchar(16) CHARACTER SET utf8	NOT NULL, DEFAULT ""	Primary key
SessionStartTime	Timestamp	NOT NULL	Primary key
SubscriptionName	varchar(31) CHARACTER SET utf8	NOT NULL	Primary key
SessionName	varchar(15) CHARACTER SET utf8	NOT NULL, DEFAULT ""	Primary key
ActivationAttributes	varchar(1023) CHARACTER SET utf8	DEFAULT NULL	Serialized activation attributes of type AttrSeq (defined in SAE external interface)

The service sessions table has a composite primary key of (UserIpAddress, VpnID, SessionStartTime, SubscriptionName, SessionName). UserIpAddress + VpnID + SessionStartTime uniquely identifies an attachment session. SubscriptionName + SessionName identifies a service session for a given attachment session.

ActivationAttributes is of varchar type that is used to store service activation attributes provided to the ASG with the request. This is the JSON (JavaScript Object Notation) serialized form of the AttrSeq attribute, which is a sequence of structured attributes.

```
struct Attr {
    string name;
    WStringSeq values;
};
```

AttrSeq serialized with JSON will be something like [{name:"name", values:["val1", ...]},...].

The service sessions table is not configurable. If modifications to the length of the UserIPAddress and VpnID fields are made in the subscriber sessions table, the SRC software will adjust the length of these fields in the service sessions table.

- Related Topics**
- Configuration Changes and Their Impact on the SSR Cluster on page 363
 - Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table (SRC CLI) on page 371
 - Configuring the Fields in the Subscriber Sessions Table (SRC CLI) on page 370
 - Modifying the SSR Database Schema in an Active Cluster (SRC CLI) on page 372

Overview of Making Modifications to the SSR Database Schema

Whenever you have modified, added, or deleted fields from the subscriber sessions table, you apply the new database schema by destroying and re-creating the SSR database.



NOTE: This is a destructive process, and must be performed during a schedule maintenance window. Destroying the database causes all data in the database to be lost.

Because the configuration of the database is stored in Juniper Networks database, modifications to the database schema can be made from any node in the SSR cluster. However, destroying and recreating the database can be performed only from a client node (with or without a management server).



NOTE: Changes to the SSR database schema requires you to restart all SRC components.

- Related Topics**
- Modifying the SSR Database Schema in an Active Cluster (SRC CLI) on page 372
 - SSR Database Schema on page 347
 - Configuring the Fields in the Subscriber Sessions Table (SRC CLI) on page 370
 - Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table (SRC CLI) on page 371
 - Creating the SSR Database (SRC CLI) on page 382
 - Deleting the SSR Database (SRC CLI) on page 382

SSR Database Operating Modes

SSR has both a running configuration and an offline configuration. In addition, the database has two modes of operation: maintenance mode and production mode. The running configuration is the configuration currently running on the SSR database. The offline configuration is used to store the configuration entered through the SRC CLI while

the SSR database is running. When you commit your changes through the SRC CLI, the configuration is written to offline configuration. In maintenance mode, database components read the offline configuration. The moment the SSR database is placed into production mode, a snapshot of the offline configuration is taken and saved in to the running configuration. In production mode, SSR database components only read the running configuration. Any configuration changes committed through the SRC CLI are stored in offline configuration and do not affect the SSR database components. Placing the SSR database into maintenance mode causes the SRC components to read from the offline configuration. The previous running configuration is dumped. The purpose of production mode is to protect SSR database components from accidental configuration changes. Maintenance mode allows you to stage configuration changes without impacting your working cluster.

- Related Topics**
- [Placing the SSR Database into Maintenance Mode \(SRC CLI\) on page 381](#)
 - [Placing the SSR Database into Production Mode \(SRC CLI\) on page 381](#)

[Distributing the SSR Cluster Configuration and Enabling SSR Client Components](#)

The SSR is a distributed system. Multiple C Series Controllers participate in the SSR cluster, and each system has a different role and hosts a different type of cluster node. Instead of configuring each C Series Controller individually by its role in the SSR cluster, the configuration is stored in the centralized Juniper Networks database. You can configure the SSR cluster from any C Series Controller. The SSR driver in a C Series Controller distributes the configuration to each node in the cluster. When you enable the SSR component in the system, the SSR driver loads the configuration from Juniper Networks database and finds the role of the system in SSR. Then it loads the relevant configuration for the specific role of the node. Finally, it starts the SSR components corresponding to the role of the node.

[Enabling, Restarting and Disabling the SSR Component Database](#)

Although the configuration is distributed to all nodes in the SSR cluster, enabling, disabling and restarting the SSR component database is a local function, which must be performed on each node in the cluster. Enabling or restarting the component database on a node starts or restarts the processes for all SSR components in the node. Disabling the component database on a node stops all processes for all SSR components in the node. After initially configuring your cluster, you must enable the component database on each node in the cluster one by one. In addition, certain configuration changes require you to disable, enable, or restart the component database on each node in the cluster one by one. Be sure to review and carefully follow the procedures described in “Configuring the Initial SSR Cluster (SRC CLI)” on page 365. When making changes to an active cluster, be sure to review each maintenance procedure beforehand and follow the steps carefully.

- Related Topics**
- [Overview of the Juniper Networks Database](#)
 - [Configuring the Initial SSR Cluster \(SRC CLI\) on page 365](#)
 - [Modifying the SSR Database Schema in an Active Cluster \(SRC CLI\) on page 372](#)
 - [Adding Data Nodes to an Active SSR Cluster \(SRC CLI\) on page 374](#)

- Adding Client Nodes to an Active SSR Cluster (SRC CLI) on page 375
- Adding a Management Server to an Active SSR Cluster on page 377
- Removing Data Nodes from an Active SSR Cluster on page 378
- Removing a Client Node from an Active SSR Cluster on page 379
- Removing a Management Server from an Active SSR Cluster on page 380

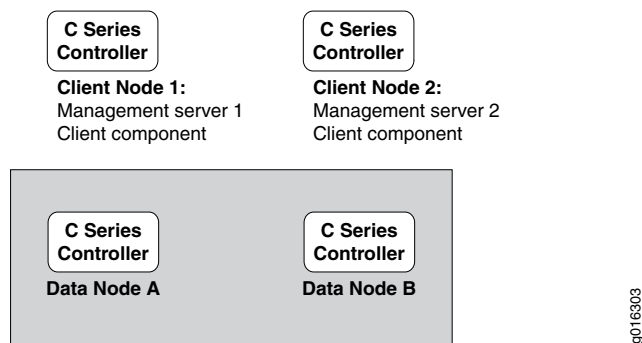
Planning Your Session State Registrar Cluster

- Planning the SSR Cluster Topology on page 355
- SSR Cluster Planning Worksheets on page 357

Planning the SSR Cluster Topology

The topology of all SSR clusters begins with the four C Series Controllers required to implement the SSR Starter Kit which licenses you for two C Series Controllers configured as client nodes, each hosting a management server and SSR client components; and two C Series Controllers, each hosting a data node. Figure 60 on page 355 shows the SSR Starter Kit basic configuration.

Figure 60: Basic SSR Starter Kit Cluster



Identifying the C Series Controllers in the SSR Cluster

Each node in the cluster is identified by its IP address. The type of node can be either a data node, a client node, or a data client node, which can be used only for demo purposes. Up to four data nodes and twenty-four client nodes can be configured. The number of data nodes configured must match the template selected with the geometry option (**two-data-node** or **four-data-node**).

- Data node—C Series Controller running an instance of the data node component.
- Client node—C Series Controller running an instance of a Juniper Networks component that is a client of the SSR (for example, SAE or SIC). In addition, each client node hosts

an instance of the database front-end component (SQL node component). Up to three client nodes may optionally be configured to run an instance of the management server component.



NOTE: All data nodes must have equal processor power, memory space, and available bandwidth because they are tightly coupled and share data. If the overall throughput of the data nodes varies from node to node, performance degrades. Therefore, all data nodes must be of the same C Series Controller model, either all C2000 model or all C4000 model.

An example of node ID assignments is shown in Table 21 on page 356. In this example, the cluster consists of two clients nodes, each hosting an SSR client component, and two data nodes.



NOTE: Because Ethernet bonding is recommended, only one IP address is used for each C Series Controller.

Table 21: Example Allocation of Node IDs

IP Address and Type	Bonded IP Address	Component	Node ID Assigned to Component
192.168.0.18 (Data node)	192.168.0.18	data component	18
192.168.0.19 (Data node)	192.168.0.19	data component	19
192.168.0.1 (Client node)	192.168.0.1	Management server component	1
		Database front-end component (SQL component)	6
		Client component	41
192.168.0.2 (Client node)	192.168.0.2	Management server component	2
		Database front-end component (SQL component)	7
		Client component	42

Related Topics

- C Series Controller Requirements on page 336
- SSR Node Types on page 334
- Supported SSR Cluster Configurations on page 341

- SSR Cluster Configurations Overview on page 337
- Scaling the SSR Cluster on page 338
- SSR Cluster Planning Worksheets on page 357

SSR Cluster Planning Worksheets

Use the worksheet shown in Table 22 on page 357 to plan your SSR Starter Kit cluster. The worksheet allows you to record IP addresses and node IDs for the four C Series Controllers. Record the node IDs after the SRC software has assigned them. If you use more nodes, see the worksheet in Table 23 on page 358 for the Expansion Kit.

Table 22: SSR Starter Kit Cluster Worksheet

Nodes and Hosted Components	Node IP Address	Bonded IP Address	Node ID Assigned to Component
Client node hosting: <ul style="list-style-type: none">• Database front-end component (SQL)• Management server component• Client component			
Client node hosting: <ul style="list-style-type: none">• Database front-end component (SQL)• Management server component• Client component			
Data node hosting: <ul style="list-style-type: none">• back-end data component			
Data node hosting: <ul style="list-style-type: none">• back-end data component			

A cluster can include either two or four data nodes, up to three management servers and up to twenty-four client nodes, each hosting SSR client components such as the SIC, SAE and so on. The worksheet in Table 23 on page 358 provides locations for you to record the IP addresses and node IDs for this maximum configuration.

Table 23: Expanded SSR Cluster Planning Worksheet

Product Name	Nodes and Hosted Components	Node IP Address	Bonded IP Address	Node ID Assigned to Component
Data Node Expansion Kit	Data node hosting: <ul style="list-style-type: none">• back-end data component			
	Data node hosting: <ul style="list-style-type: none">• back-end data component			
Management Server Expansion Kit	Management server component			

Table 23: Expanded SSR Cluster Planning Worksheet *(continued)*

Product Name	Nodes and Hosted Components	Node IP Address	Bonded IP Address	Node ID Assigned to Component
Additional client nodes	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			

Table 23: Expanded SSR Cluster Planning Worksheet (*continued*)

Product Name	Nodes and Hosted Components	Node IP Address	Bonded IP Address	Node ID Assigned to Component
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			

Table 23: Expanded SSR Cluster Planning Worksheet *(continued)*

Product Name	Nodes and Hosted Components	Node IP Address	Bonded IP Address	Node ID Assigned to Component
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			

- Related Topics
- [SSR Node Types on page 334](#)
 - [Supported SSR Cluster Configurations on page 341](#)
 - [SSR Cluster Configurations Overview on page 337](#)
 - [Scaling the SSR Cluster on page 338](#)
 - [Planning the SSR Cluster Topology on page 355](#)

Configuring the Session State Registrar (SRC CLI)

- Configuration Changes and Their Impact on the SSR Cluster on page 363
- Configuration Statements for the SSR Cluster on page 364
- Configuring the Initial SSR Cluster (SRC CLI) on page 365
- Configuring the SSR Cluster ID (SRC CLI) on page 366
- Configuring the SSR Cluster Geometry (SRC CLI) on page 367
- Configuring the Nodes in the SSR Cluster (SRC CLI) on page 368
- Configuring the Management Servers in the SSR Cluster (SRC CLI) on page 369
- Configuring the Fields in the Subscriber Sessions Table (SRC CLI) on page 370
- Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table (SRC CLI) on page 371
- Modifying the SSR Database Schema in an Active Cluster (SRC CLI) on page 372
- Modifying Attribute Mapping in an Active SSR Cluster (SRC CLI) on page 373
- Adding Data Nodes to an Active SSR Cluster (SRC CLI) on page 374
- Adding Client Nodes to an Active SSR Cluster (SRC CLI) on page 375
- Adding a Management Server to an Active SSR Cluster on page 377
- Removing Data Nodes from an Active SSR Cluster on page 378
- Removing a Client Node from an Active SSR Cluster on page 379
- Removing a Management Server from an Active SSR Cluster on page 380

Configuration Changes and Their Impact on the SSR Cluster

Certain configuration changes made on an active SSR cluster may require you to shut down and reinitialize the cluster—for example, when the cluster geometry is changed or when modifications are made to the database schema. Here are some examples of configuration changes and their impact on the SSR cluster:

- Dropping or adding of columns in the SSR database schema requires you to re-create the database by executing the **request database delete database** and then **request database create database** commands. It does not require shutdown of the cluster data nodes.



NOTE: Changes to the SSR database schema requires you to restart all SRC components.

- Moving from a two-node geometry to a four-node geometry, or vice versa, requires restarting of all management servers and data nodes in the SSR cluster by executing the **restart component database** command on each of the data nodes and management servers
- Adding or removing client nodes requires restarting all management servers and the new client node (or shutdown the removed client node) in the SSR cluster by executing the **restart component database** command on each management server and the new client node.

To facilitate SSR maintenance, a command is used to set the SSR database into maintenance mode or production mode.

Related Topics

- Placing the SSR Database into Maintenance Mode (SRC CLI) on page 381
- Modifying the SSR Database Schema in an Active Cluster (SRC CLI) on page 372
- Modifying Attribute Mapping in an Active SSR Cluster (SRC CLI) on page 373
- Adding Data Nodes to an Active SSR Cluster (SRC CLI) on page 374
- Adding Client Nodes to an Active SSR Cluster (SRC CLI) on page 375
- Adding a Management Server to an Active SSR Cluster on page 377

Configuration Statements for the SSR Cluster

Use the following statements to configure the SSR cluster at the **[edit]** hierarchy level:

```
shared database cluster {
    primary;
}
shared database cluster primary nodes {
    geometry [(all-in-one | two-data-node | four-data-node)];
}
shared database cluster primary nodes {
    node address address;
    platform [(C2000 | C4000)];
    type [(data-node | client-node | data-client-node)];
}
shared database cluster primary nodes node address client-node {
    management-server;
}
shared database cluster primary schema table subscriber-sessions field name {
    type [(int | string | binary)];
    size size;
    require-value
    indexed;
    default default;
    variable-length;
```

```

}
shared database cluster primary attribute-associations table name field name name {
    sae-plugin-attribute sae-plugin-attribute ;
}

```

- Related Topics**
- Configuring the Initial SSR Cluster (SRC CLI) on page 365
 - Overview of the Session State Registrar on page 333
 - Configuration Changes and Their Impact on the SSR Cluster on page 363

Configuring the Initial SSR Cluster (SRC CLI)

Configuring your initial cluster is a multistep process that requires you to install your C Series Controllers, load the SRC software, and perform the following configuration steps.

For information about supported cluster configurations see “Supported SSR Cluster Configurations” on page 341.



NOTE: This procedure assumes that you are configuring the cluster for the first time and as such, the SSR database is already in maintenance mode.

To configure your initial SSR cluster:

1. Configure the cluster ID.
See “Configuring the SSR Cluster ID (SRC CLI)” on page 366.
2. Configure the cluster geometry.
See “Configuring the SSR Cluster Geometry (SRC CLI)” on page 367.
3. Configure the role of each node in the cluster
See “Configuring the Nodes in the SSR Cluster (SRC CLI)” on page 368.
4. Configure the management servers.
See “Configuring the Management Servers in the SSR Cluster (SRC CLI)” on page 369.
5. (Optional) Make desired changes to the default subscriber sessions table.
See “Configuring the Fields in the Subscriber Sessions Table (SRC CLI)” on page 370.
6. (Optional) Make desired changes to the default attribute associations table.
See “Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table (SRC CLI)” on page 371.
7. Commit the configuration.
See Committing a Configuration and Exiting Configuration Mode.
8. Enable the component database (must be done on each node in the SSR cluster.)
See “Enabling the SSR Database (SRC CLI)” on page 383.

9. Create the SSR database. (There is no need to destroy the database because it does not exist at initial startup).
See “Creating the SSR Database (SRC CLI)” on page 382.
10. Enable each SSR client component.
See Enabling SRC Components.
11. Verify status of the cluster.
See “Viewing the Status of the SSR Cluster (SRC CLI)” on page 385.
12. Place the cluster into production mode.
See “Placing the SSR Database into Production Mode (SRC CLI)” on page 381.

- Related Topics**
- Configuration Statements for the SSR Cluster on page 364
 - Overview of the Session State Registrar on page 333
 - Configuration Changes and Their Impact on the SSR Cluster on page 363
 - SSR Node Types on page 334
 - SSR Database Schema on page 347
 - SSR Client Component Attribute Associations

Configuring the SSR Cluster ID (SRC CLI)

Use the following configuration statements to configure the cluster ID:

```
shared database cluster {  
    primary;  
}
```

Configure the cluster ID from any node in the cluster:



NOTE: In this release of SRC software, the cluster ID is fixed to *primary* and cannot be modified. Only one cluster is supported.

1. Set the cluster ID:
`user@host# edit shared database cluster primary`

- Related Topics**
- Planning the SSR Cluster Topology on page 355
 - SSR Cluster Planning Worksheets on page 357

Configuring the SSR Cluster Geometry (SRC CLI)

The cluster geometry specifies the number of data nodes in the cluster. In a two-data-node geometry, the two data nodes are configured in the same node group, which means they are backups for each other. In a four-data-node geometry, four data nodes are configured in two node groups. Each node group hosts 50% of the data and contains two data node servers. The two data nodes in each group are backups for each other.



NOTE: All data nodes must have equal processor power, memory space, and available bandwidth because they are tightly coupled and share data. If the overall throughput of the data nodes varies from node to node, performance degrades. Therefore, all data nodes must be of the same C Series Controller model, either all C2000 model or all C4000 model.

Use the following configuration statements to configure the cluster geometry:

```
shared database cluster primary nodes {
    geometry [(all-in-one | two-data-node | four-data-node)];
}
```

Configure the cluster geometry from any node in the cluster:

1. From configuration mode enter the statement to configure the cluster geometry:

```
user@host# edit shared database cluster primary nodes
```

2. Specify the cluster geometry. For example, to set the geometry for two data node:

```
[edit shared database cluster primary nodes]
user@host# set geometry two-data-node
```

The geometry can be set to either **two-data-node**, **four-data-node**, or **all-in-one**. In two-data-node geometry, the two data nodes are configured in the same node group, which means they are backups for each other. In four-data-node, four data nodes are configured in two node groups. Each node group hosts 50% of the data and contains two data node servers. The two data nodes in each group are backups for each other. The **all-in-one** setting is for demonstration purposes only, and requires the node type to be configured as a **data-client-node**.

Related Topics

- [SSR Cluster Configurations Overview on page 337](#)
- [SSR Cluster Network Requirements on page 339](#)
- [SSR Node Groups on page 335](#)
- [Scaling the SSR Cluster on page 338](#)
- [Supported SSR Cluster Configurations on page 341](#)

Configuring the Nodes in the SSR Cluster (SRC CLI)

The node configuration is a list of all nodes in the cluster. Each entry in the list declares either a data node or a client node. For client nodes, an optional keyword specifies that a management server is enabled on the node.

The SSR database contains a node collection. Each node is identified by its IP address. The type of node can be set to either **data-node**, **client-node** or **data-client-node**, which can only be used when the cluster geometry is set to **all-in-one**. If the node type is set to **client-node** or **data-client-node**, a client node object appears under the node for setting the client node related options. You can configure up to four data nodes and twenty-four client nodes.



NOTE: The number of data nodes configured must match the template selected with the geometry setting.



NOTE: All data nodes must have equal processor power, memory space, and available bandwidth because they are tightly coupled and share data. If the overall throughput of the data nodes varies from node to node, performance degrades. Therefore, all data nodes must be of the same C Series Controller model, either all C2000 model or all C4000 model.

Use the following configuration statements to configure at least one client node and two data nodes in the cluster:

```
shared database cluster primary nodes{
  node address address;
  platform [(C2000 | C4000)];
  type [(data-node | client-node | data-client-node)];
}
```

To configure each node in the cluster (perform steps from any node in the cluster and repeat for each node in the SSR cluster):

1. From configuration mode, access the statement to configure the cluster nodes.

```
user@host# edit shared database cluster primary nodes node
```

2. Specify the node's IP address.

```
[edit shared database cluster primary nodes node]
user@host# set address address
```

3. Specify the platform type for the node.

```
[edit shared database cluster primary nodes node]
user@host# set platform [(C2000 | C4000)]
```

4. Configure the node type.


```
[edit shared database cluster primary nodes node]
user@host# set type [(data-node | client-node | data-client-node)]
```

where the type is one of the following values:

- **data-node**—Configures the node as a data node. Data nodes are always configured in pairs, and your cluster can contain either two or four data nodes.
- **client-node**—Configures the node as a client node. Client nodes can optionally host a management server.
- **data-client-node**—For demonstration purposes only. When this option is selected, the node contains a **data-node** component, a **client-node** component, and a management server. This setting is only used when the cluster geometry is set to **all-in-one**.

- Related Topics**
- SSR Node Types on page 334
 - C Series Controller Requirements on page 336
 - SSR Cluster Configurations Overview on page 337
 - Scaling the SSR Cluster on page 338
 - Supported SSR Cluster Configurations on page 341
 - Configuring the Management Servers in the SSR Cluster (SRC CLI) on page 369

Configuring the Management Servers in the SSR Cluster (SRC CLI)

At least one client node must host the management server process. For redundancy at least two client nodes must be configured to host management server processes. Each management server must run on a separate client node.

Use the following configuration statement to configure the management server on a client node:

```
shared database cluster (primary) nodes node address client-node {
    management-server;
}
```

To configure the management server, perform these steps from any node in cluster and repeat steps for each client node hosting a management server:

1. From configuration mode, access the statement to configure at least one management server, and specify the IP address of the client node you want to host the management server.

```
user@host# edit shared database cluster primary nodes node address client-node
```

2. Configure the client node to host a management server.

```
[edit shared database cluster primary nodes node address client-node]
user@host# set management-server
```

- Related Topics**
- [SSR Node Types on page 334](#)
 - [Configuring the Nodes in the SSR Cluster \(SRC CLI\) on page 368](#)

Configuring the Fields in the Subscriber Sessions Table (SRC CLI)

The format of subscriber identity information stored for each session is managed by the subscriber sessions table, which is controlled by the cluster's client node/management servers. The configuration of the subscriber sessions table is copied to all nodes in the cluster, so all nodes operate with the same information.

The preconfigured schema defines the table's indexes. The default set of database fields in the subscriber sessions table addresses the needs of most carriers, however you can modify certain fields to address unique needs and situations. "SSR Database Schema" on page 347 describes the default configuration and fields that can be modified.



NOTE: Changes to the SSR database schema requires you to restart all SRC components.

Use the following configuration statements to modify the fields in the subscriber sessions table:

```
shared database cluster primary schema table subscriber-sessions field name {  
  type [(int | string | binary)];  
  size size;  
  require-value;  
  indexed;  
  default default;  
  variable-length;  
}
```

To configure the subscriber sessions table:

1. (Optional) From configuration mode, access the statement to configure the fields in the subscriber sessions table. For example, to add a new field called DeviceName:

```
user@host# edit shared database cluster primary schema table subscriber-sessions  
field name
```

2. (Optional) Specify the field type. Values for type correspond to legal SQL data types. For example, to specify the new field as a text string:

```
[edit shared database cluster primary schema table subscriber-sessions field name]  
user@host# set type string
```

3. (Optional) Specify the field size in bytes. For example, to specify the field size as two bytes:

```
[edit shared database cluster primary schema table subscriber-sessions field name]  
user@host# set size 2
```

4. (Optional) Specify whether null values are allowed for the value of the field. For example, to specify that a value is required:

```
[edit shared database cluster primary schema table subscriber-sessions field name]
user@host# set require-value
```

5. (Optional) Specify whether you want the table to be indexed by this field. For example, to specify that the table is indexed by this field:

```
[edit shared database cluster primary schema table subscriber-sessions field name]
user@host# set indexed
```

6. (Optional) Specify the default value for the field corresponding to the type. For example, to specify the default as `nas123`:

```
[edit shared database cluster primary schema table subscriber-sessions field name]
user@host# set default nas123
```

7. (Optional) If the type is specified as either binary or string, specify whether the value is variable. If set, the SQL data type will be varbinary or varchar, respectively. For example, to specify the type as variable:

```
[edit shared database cluster primary schema table subscriber-sessions field name]
user@host# set variable-length
```

Related Topics

- SSR Database Schema on page 347
- Overview of Making Modifications to the SSR Database Schema on page 351
- Creating the SSR Database (SRC CLI) on page 382
- Deleting the SSR Database (SRC CLI) on page 382

Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table (SRC CLI)



NOTE: Any fields in the subscriber sessions table that have a “not null” requirement must be mapped to either a request attribute or variable.

Use the following configuration statements to map an attribute in the subscriber sessions table to an SAE plugin attribute:

```
shared database cluster (primary) attribute-associations entity name
  shared database cluster primary attribute-associations table name field name {
    sae-plugin-attribute sae-plugin-attribute ;
  }
```

The following procedure can be performed from any node.

To map an attribute in the subscriber sessions table to an SAE plugin attribute:

1. (Optional) From configuration mode, access the configuration statement that configures the name of the table in the SSR database to which you want to make

an association. For example, to map the subscriber sessions table attribute VpnID to the SAE plugin attribute vpn-id:

```
user@host# edit shared database cluster primary attribute-associations entity  
subscriber-sessions
```

2. Specify the name of the field in the subscriber sessions table and the SAE plugin attribute.

```
[edit shared database cluster primary attribute-associations entity subscriber-sessions]  
user@host# set field VpnID sae-plugin-attribute vpn-id
```

3. Commit the configuration.

```
user@host# commit
```

- Related Topics**
- SSR Client Component Attribute Associations
 - SSR Database Schema on page 347
 - Configuring the Fields in the Subscriber Sessions Table (SRC CLI) on page 370

Modifying the SSR Database Schema in an Active Cluster (SRC CLI)

The default database schema is sufficient for most carrier environments. However, if you need to modify the default schema, you need to modify the fields in the subscriber sessions table, re-create the database, and apply the new schema. See “SSR Database Schema” on page 347 for a description of the default database schema and which fields can be modified.



CAUTION: The following procedure requires you to re-create the SSR database. This is a destructive process that deletes all information in the database. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding. In addition, because the steps required to re-create the SSR database can be performed only from a client node, we recommend that you perform this procedure from a client node.



NOTE: This procedure assumes that you are working with an active cluster and that the SSR database is in production mode. If you are unsure what mode the database is in, see “Viewing the SSR Database Mode (SRC CLI)” on page 385.



NOTE: Changes to the SSR database schema requires you to restart all SRC components.

To modify the SSR database schema in an active cluster:

1. Modify the fields in the subscriber sessions table.
See “Configuring the Fields in the Subscriber Sessions Table (SRC CLI)” on page 370.
2. Commit the configuration.
See Committing a Configuration and Exiting Configuration Mode.
3. Place the database into maintenance mode.
See “Placing the SSR Database into Maintenance Mode (SRC CLI)” on page 381.
4. Delete the existing database.
See “Deleting the SSR Database (SRC CLI)” on page 382.
5. Create the database with the modified fields.
See “Creating the SSR Database (SRC CLI)” on page 382.
6. Verify the cluster configuration changes by viewing the status of the database.
See “Viewing the Status of the SSR Cluster (SRC CLI)” on page 385.
7. Restart all SSR client components in the cluster.
See Enabling SRC Components.
8. Place the new database into production mode.
See “Placing the SSR Database into Production Mode (SRC CLI)” on page 381.

- Related Topics**
- SSR Database Schema on page 347
 - Configuration Changes and Their Impact on the SSR Cluster on page 363
 - Overview of Making Modifications to the SSR Database Schema on page 351
 - SSR Client Component Attribute Associations

Modifying Attribute Mapping in an Active SSR Cluster (SRC CLI)



NOTE: This procedure assumes that you are working with an active cluster and that the SSR database is in production mode. If you are unsure what mode the database is in, see “Viewing the SSR Database Mode (SRC CLI)” on page 385.



NOTE: Any fields in the subscriber sessions table that have a “not null” requirement must be mapped to either a request attribute or variable.

This procedure can be performed from any node in the SSR cluster. To modify the SSR client component attribute associations in an active cluster:

1. Modify the attribute associations.
See “Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table (SRC CLI)” on page 371.
2. Commit the configuration.
See Committing a Configuration and Exiting Configuration Mode.
3. Place the database into maintenance mode.
See “Placing the SSR Database into Maintenance Mode (SRC CLI)” on page 381.
4. Verify the cluster configuration changes by viewing the status of the database.
See “Viewing the Status of the SSR Cluster (SRC CLI)” on page 385.
5. Place the database into production mode.
See “Placing the SSR Database into Production Mode (SRC CLI)” on page 381.

- Related Topics**
- SSR Database Schema on page 347
 - Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table (SRC CLI) on page 371

Adding Data Nodes to an Active SSR Cluster (SRC CLI)



NOTE: This procedure makes the following assumptions:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see “Viewing the SSR Database Mode (SRC CLI)” on page 385
- Your SSR cluster currently only has two data nodes.
- You have physically installed the two additional C Series Controllers that will be acting as data nodes, made the interface connections to the cluster, and installed the SRC software.



CAUTION: Changing the cluster geometry causes the SSR database to be destroyed. All data will be lost and you need to re-create the database. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

To add data nodes to an active cluster:

1. Configure each of the C Series Controllers as data nodes.
See “Configuring the Nodes in the SSR Cluster (SRC CLI)” on page 368.
2. Change the SSR cluster geometry to *four-data-node*.

See “Configuring the SSR Cluster Geometry (SRC CLI)” on page 367.

3. Commit the configuration.

See Committing a Configuration and Exiting Configuration Mode.

4. Place the database into maintenance mode.

See “Placing the SSR Database into Maintenance Mode (SRC CLI)” on page 381.

5. Disable the component database on each management server in the cluster (one by one).

See “Disabling the SSR Database (SRC CLI)” on page 383.

6. Enable the component database on each management server in the cluster (one by one).

See “Enabling the SSR Database (SRC CLI)” on page 383.

7. Restart the component database on each of the existing data nodes (one by one).

See “Restarting the SSR Database (SRC CLI)” on page 383.

8. Enable the component database on each of the new data nodes (one by one).

See “Enabling the SSR Database (SRC CLI)” on page 383.

9. Verify the cluster configuration changes by viewing the status of the database.

See “Viewing the Status of the SSR Cluster (SRC CLI)” on page 385.

10. re-create the SSR database.

See “Creating the SSR Database (SRC CLI)” on page 382.

11. Restart all SSR client components.

See Enabling SRC Components

12. Place the database into production mode.

See “Placing the SSR Database into Production Mode (SRC CLI)” on page 381.

Adding Client Nodes to an Active SSR Cluster (SRC CLI)



NOTE: This procedure makes the following assumptions:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see “Viewing the SSR Database Mode (SRC CLI)” on page 385.
- You have physically installed the new C Series Controllers that will be acting as a client node, made the interface connections to the cluster, and installed the SRC software.



.....

CAUTION: This procedure requires you to restart the management servers in the cluster, which is a disruptive process. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

.....

To add a client node to an active cluster:

1. Configure the C Series Controller as a client node.
See "Configuring the Nodes in the SSR Cluster (SRC CLI)" on page 368.
2. (Optional) Configure the new client node to host a management server.
See "Configuring the Management Servers in the SSR Cluster (SRC CLI)" on page 369.
3. Commit the configuration.
See Committing a Configuration and Exiting Configuration Mode.
4. Place the database into maintenance mode.
See "Placing the SSR Database into Maintenance Mode (SRC CLI)" on page 381.
5. Disable the component database on each management server in the cluster (one by one).
See "Disabling the SSR Database (SRC CLI)" on page 383.
6. Enable the component database on each management server in the cluster (one by one).
See "Enabling the SSR Database (SRC CLI)" on page 383.
7. (Optional) If the new client node is hosting a management server, you need to restart the component database and all SSR client components on all client nodes in the cluster (one by one). If the new client node is not hosting a management server, this step is not necessary.
See "Restarting the SSR Database (SRC CLI)" on page 383.
See Enabling SRC Components.
8. Enable the component database on the new client node. If you have added more than one client node, perform this step on each new client node (one by one).
See "Enabling the SSR Database (SRC CLI)" on page 383.
9. Verify the cluster configuration changes by viewing the status of the database.
See "Viewing the Status of the SSR Cluster (SRC CLI)" on page 385.
10. Place the database into production mode.
See "Placing the SSR Database into Production Mode (SRC CLI)" on page 381.

Adding a Management Server to an Active SSR Cluster



NOTE: This procedure makes the following assumptions:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see “Viewing the SSR Database Mode (SRC CLI)” on page 385.
- The client node is already installed and configured. If you also need to install the client node, use the procedure for “Adding Client Nodes to an Active SSR Cluster (SRC CLI)” on page 375.



CAUTION: This procedure requires you to restart the component database on all client nodes and management servers in the cluster, which is a disruptive process. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

To add a management server to a client node in an active cluster:

1. Select the client node you want to host the new management server, and configure the management server.
See “Configuring the Management Servers in the SSR Cluster (SRC CLI)” on page 369.
2. Commit the configuration.
See Committing a Configuration and Exiting Configuration Mode.
3. Place the database into maintenance mode.
See “Placing the SSR Database into Maintenance Mode (SRC CLI)” on page 381.
4. Disable the component database on all management servers in the cluster (one by one).
See “Disabling the SSR Database (SRC CLI)” on page 383.
5. Enable the component database on all management servers in the cluster (one by one).
See “Enabling the SSR Database (SRC CLI)” on page 383.
6. Restart the component database and all SSR client components on all client nodes in the cluster (one by one).
See “Restarting the SSR Database (SRC CLI)” on page 383.
See Enabling SRC Components.
7. Verify the cluster configuration changes by viewing the status of the database.
See “Viewing the Status of the SSR Cluster (SRC CLI)” on page 385.
8. Place the database into production mode.

See “Placing the SSR Database into Production Mode (SRC CLI)” on page 381.

Removing Data Nodes from an Active SSR Cluster



NOTE: This procedure makes the following assumptions:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see “Viewing the SSR Database Mode (SRC CLI)” on page 385.
- Your cluster currently contains four data nodes and you want to change to a two-data-node cluster. Data nodes can be added or removed only in pairs.



CAUTION: Changing the SSR cluster geometry (adding or removing data nodes) destroys all data in the SSR database. All data is lost. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

To remove two data nodes from an active cluster:

1. Reconfigure the cluster topology to a two-data-node geometry.
See “Configuring the SSR Cluster Geometry (SRC CLI)” on page 367.
2. Commit the configuration.
See Committing a Configuration and Exiting Configuration Mode.
3. Place the database into maintenance mode.
See “Placing the SSR Database into Maintenance Mode (SRC CLI)” on page 381.
4. Disable the component database on all management servers in the cluster (one by one).
See “Disabling the SSR Database (SRC CLI)” on page 383.
5. Disable the component database on the two data nodes you removed from the cluster (one by one).
See “Disabling the SSR Database (SRC CLI)” on page 383.
6. Enable the component database on all management servers in the cluster (one by one).
See “Enabling the SSR Database (SRC CLI)” on page 383.
7. Restart the component database on the two remaining data nodes in the cluster (one by one).
See “Restarting the SSR Database (SRC CLI)” on page 383.
8. re-create the SSR database.
See “Creating the SSR Database (SRC CLI)” on page 382.

9. Restart all SSR client components.
See Enabling SRC Components
10. Verify the cluster configuration changes by viewing the status of the database.
See “Viewing the Status of the SSR Cluster (SRC CLI)” on page 385.
11. Place the database into production mode.
See “Placing the SSR Database into Production Mode (SRC CLI)” on page 381.

Removing a Client Node from an Active SSR Cluster



NOTE: This procedure makes the following assumptions:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see “Viewing the SSR Database Mode (SRC CLI)” on page 385.



CAUTION: This procedure requires you to restart the component database on all management servers in the cluster, which is a disruptive process. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

To remove a client node from an active cluster:

1. Place the database into maintenance mode.
See “Placing the SSR Database into Maintenance Mode (SRC CLI)” on page 381.
2. Disable the component database on all management servers in the cluster (one by one).
See “Disabling the SSR Database (SRC CLI)” on page 383.
3. Disable the component database on the client node you removed from the cluster.
See “Disabling the SSR Database (SRC CLI)” on page 383.
4. Enable the component database on all management servers in the cluster (one by one).
See “Enabling the SSR Database (SRC CLI)” on page 383.
5. Restart the SSR component database on all client nodes.
See “Enabling the SSR Database (SRC CLI)” on page 383.
6. Restart all SSR client components.
See Enabling SRC Components.
7. Verify the cluster configuration changes by viewing the status of the database.

See “Viewing the Status of the SSR Cluster (SRC CLI)” on page 385.

8. Place the database into production mode.

See “Placing the SSR Database into Production Mode (SRC CLI)” on page 381.

Removing a Management Server from an Active SSR Cluster



NOTE: This procedure makes the following assumptions:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see “Viewing the SSR Database Mode (SRC CLI)” on page 385.
- The client node that is hosting the management server is installed, configured and active, and you want it to remain in the cluster as a client node only.



CAUTION: This procedure requires you to restart the component database on all management servers in the cluster, which is a disruptive process. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

To remove a management server from a client node in an active cluster:

1. Select the client node from which you want to remove the management server, and reconfigure the node as a client node without a management server.
See “Configuring the Nodes in the SSR Cluster (SRC CLI)” on page 368.
2. Commit the configuration.
See Committing a Configuration and Exiting Configuration Mode.
3. Place the database into maintenance mode.
See “Placing the SSR Database into Maintenance Mode (SRC CLI)” on page 381.
4. Disable the component database on all management servers in the cluster (one by one).
See “Disabling the SSR Database (SRC CLI)” on page 383.
5. Enable the component database on all management servers in the cluster (one by one).
See “Enabling the SSR Database (SRC CLI)” on page 383.
6. Verify the cluster configuration changes by viewing the status of the database.
See “Viewing the Status of the SSR Cluster (SRC CLI)” on page 385.
7. Place the database into production mode.
See “Placing the SSR Database into Production Mode (SRC CLI)” on page 381.

CHAPTER 29

Managing the SSR Cluster

- Placing the SSR Database into Maintenance Mode (SRC CLI) on page 381
- Placing the SSR Database into Production Mode (SRC CLI) on page 381
- Deleting the SSR Database (SRC CLI) on page 382
- Creating the SSR Database (SRC CLI) on page 382
- Enabling the SSR Database (SRC CLI) on page 383
- Disabling the SSR Database (SRC CLI) on page 383
- Restarting the SSR Database (SRC CLI) on page 383
- Deleting All Subscriber Sessions in the SSR Database on page 384
- Deleting Subscriber Sessions in the SSR Database By IP Address on page 384

Placing the SSR Database into Maintenance Mode (SRC CLI)

- To place the SSR database into maintenance mode, from operational mode:
`user@host> request database enter maintenance-mode`

Related Topics

- SSR Database Operating Modes on page 351
- Placing the SSR Database into Production Mode (SRC CLI) on page 381

Placing the SSR Database into Production Mode (SRC CLI)

- To place the SSR database into production mode, from operational mode:
`user@host> request database enter production-mode`

Related Topics

- SSR Database Operating Modes on page 351
- Placing the SSR Database into Maintenance Mode (SRC CLI) on page 381

Deleting the SSR Database (SRC CLI)



CAUTION: This command destroys the SSR database and all its contents and should be performed only during a maintenance window.



NOTE: This command must be executed from a client node and can be executed only when the database is in maintenance mode.

- From operational mode, destroy the database.

```
user@host> request database delete database
```

Related Topics

- SSR Database Schema on page 347
- Overview of Making Modifications to the SSR Database Schema on page 351
- Creating the SSR Database (SRC CLI) on page 382

Creating the SSR Database (SRC CLI)

This procedure generates the database schema using the current configuration and then creates the SSR database and its tables.



NOTE: All cluster components must have been enabled before you create the database.



NOTE: This command must be executed from a client node and can be executed only when the database is in maintenance mode.

- From operational mode, re-create the database using the modified schema.

```
user@host> request database create database
```

Related Topics

- Enabling the SSR Database (SRC CLI) on page 383
- Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 352
- SSR Database Schema on page 347
- Overview of Making Modifications to the SSR Database Schema on page 351
- SSR Database Schema on page 347

Enabling the SSR Database (SRC CLI)

This command starts all SSR component processes on the local node. When multiple nodes need to be enabled, you must execute this command on each node, one by one.



NOTE: You must execute this command on each node in the SSR cluster.

- From operational mode, enable the SSR component database.

```
user@host> enable component database
```

- Related Topics**
- Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 352
 - Overview of the Juniper Networks Database
 - Creating the SSR Database (SRC CLI) on page 382
 - Disabling the SSR Database (SRC CLI) on page 383
 - Restarting the SSR Database (SRC CLI) on page 383

Disabling the SSR Database (SRC CLI)

This command stops all SSR component processes on the local node. When multiple nodes need to be shut down, you must execute this command on each node one by one.

- From operational mode, disable the SSR component database.

```
user@host> disable component database
```

- Related Topics**
- Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 352
 - Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 352
 - Enabling the SSR Database (SRC CLI) on page 383
 - Restarting the SSR Database (SRC CLI) on page 383

Restarting the SSR Database (SRC CLI)

This command restarts all SSR component processes on the local node. When you need to restart multiple nodes, it must be executed on each node, one by one.

- From operational mode, restart the SSR component database.

```
user@host> restart component database
```

- Related Topics**
- Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 352
 - Enabling the SSR Database (SRC CLI) on page 383
 - Disabling the SSR Database (SRC CLI) on page 383

Deleting All Subscriber Sessions in the SSR Database

- From operational mode, delete all subscriber sessions and associated service sessions.
`user@host> request database delete sessions all`

- Related Topics**
- Deleting Subscriber Sessions in the SSR Database By IP Address on page 384
 - Viewing All Subscriber Sessions in the SSR Database (SRC CLI) on page 387
 - Viewing Subscriber Sessions in the SSR Database by IP Address (SRC CLI) on page 388
 - Viewing Subscriber Sessions in the SSR Database by Indexed Field (SRC CLI) on page 389

Deleting Subscriber Sessions in the SSR Database By IP Address

1. To delete subscriber sessions based on IP address, from operational mode:
`user@host>request database delete subscriber-sessions by-address start-address
start-address end-address end-address vpn-id vpn-id`
2. Enter the starting address of the IP range. If the end address is not specified, only the session matching this address is deleted.
3. (Optional) Enter the ending address of the IP range. If not specified, only the session matching the start address is deleted.
4. (Optional) Enter the VPN ID that the sessions belong to. If not specified, only sessions with public IP addresses are displayed. If value is either " " or ' ', display sessions from public network or any VPN.

- Related Topics**
- Deleting All Subscriber Sessions in the SSR Database on page 384
 - Viewing All Subscriber Sessions in the SSR Database (SRC CLI) on page 387
 - Viewing Subscriber Sessions in the SSR Database by IP Address (SRC CLI) on page 388
 - Viewing Subscriber Sessions in the SSR Database by Indexed Field (SRC CLI) on page 389

Monitoring the SSR Cluster

- Viewing the SSR Database Mode (SRC CLI) on page 385
- Viewing the Status of the SSR Cluster (SRC CLI) on page 385
- Viewing the Running Configuration of the SSR Database (SRC CLI) on page 386
- Viewing All Subscriber Sessions in the SSR Database (SRC CLI) on page 387
- Viewing Subscriber Sessions in the SSR Database by IP Address (SRC CLI) on page 388
- Viewing Subscriber Sessions in the SSR Database by Indexed Field (SRC CLI) on page 389
- Viewing the Total Number of Subscriber Sessions in the SSR Database (SRC CLI) on page 390

Viewing the SSR Database Mode (SRC CLI)

Purpose View the current operating mode of the SSR database (maintenance or production mode). This command can be executed from any node in the cluster.

Action user@host> **show database mode**
Database is in maintenance-mode

- Related Topics**
- Placing the SSR Database into Maintenance Mode (SRC CLI) on page 381
 - Placing the SSR Database into Production Mode (SRC CLI) on page 381

Viewing the Status of the SSR Cluster (SRC CLI)

Purpose View the status of the SSR database. The IP address and the connection status for each node in the cluster is displayed. This command displays all configured data nodes and management servers, regardless of whether they are connected. However, only client nodes that are connected are displayed.



NOTE: This command must be issued on a client node with a management server.

To interpret the status, make sure that you have a good understanding of the SSR concepts and terminology.

See “Overview of the Session State Registrar” on page 333 and “SSR Node Types” on page 334.

Action user@host> **show database status**

```
Data Nodes
Data Node
Address 10.227.6.49
Node ID 1
Connected Yes
Status Nodegroup: 0, Master
```

```
Management Servers
Management Server
Address 10.227.6.49
Node ID 16
Connected Yes
```

```
Connected Client Nodes
Client Node
Address 10.227.6.49
Node ID 17
Component Name mysqld
```

```
Client Node
Address 10.227.6.49
Node ID 21
Component Name DSA
```

- Related Topics**
- [SSR Node Types on page 334](#)
 - [SSR Cluster Configurations Overview on page 337](#)
 - [SSR Node Groups on page 335](#)
 - [Supported SSR Cluster Configurations on page 341](#)
 - [SSR Database Schema on page 347](#)

Viewing the Running Configuration of the SSR Database (SRC CLI)

Purpose View the database running configuration. When SSR database is in production mode, displays configuration running on SSR database. When SSR database is placed in maintenance mode, the running configuration is discarded and the SSR database running configuration is the configuration you entered in the SRC CLI. This command can be executed only on management node.

Action user@host> **show database running-configuration**

```
Data Nodes
Data Node
Address 10.227.6.49
Node ID 1
Connected Yes
Status Nodegroup: 0, Master
```

```
Management Servers
Management Server
```

```

Address  10.227.6.49
Node ID  16
Connected Yes

```

Connected Client Nodes

```

Client Node
Address      10.227.6.49
Node ID      17
Component Name mysql

```

```

Client Node
Address      10.227.6.49
Node ID      21
Component Name DSA

```

- Related Topics**
- [SSR Node Types on page 334](#)
 - [SSR Cluster Configurations Overview on page 337](#)
 - [SSR Node Groups on page 335](#)
 - [Supported SSR Cluster Configurations on page 341](#)
 - [SSR Database Schema on page 347](#)

Viewing All Subscriber Sessions in the SSR Database (SRC CLI)

Purpose View all the subscriber sessions and associated service sessions stored in the SSR database. The command only shows attributes that are not null.



NOTE: This command must be executed on a client node with or without a management server.

Enter the maximum number of sessions to display. By default the maximum number of sessions displayed is 25.

Action `user@host> show database subscriber-sessions maximum-results maximum-result`

```

UserIPAddress  0.0.0.1
VpnID
SessionStartTime 2000-01-01 00:00:00.0
UserName      abc

```

```

Service session
Subscription Name test1
SessionName      DEFAULT

```

```

Subscriber session
UserIPAddress  0.0.0.2
VpnID
SessionStartTime 2000-01-02 00:00:00.0
UserName      abc

```

```

Service session
Subscription Name test1

```

```

SessionName      DEFAULT

Subscriber session
UserIPAddress    0.0.0.3
VpnID
SessionStartTime 2000-01-03 00:00:00.0
UserName         abc

Subscriber session
UserIPAddress    0.0.0.4
VpnID
SessionStartTime 2000-01-04 00:00:00.0
UserName         abc

```

- Related Topics**
- Viewing Subscriber Sessions in the SSR Database by IP Address (SRC CLI) on page 388
 - Viewing Subscriber Sessions in the SSR Database by Indexed Field (SRC CLI) on page 389
 - Viewing the Total Number of Subscriber Sessions in the SSR Database (SRC CLI) on page 390

Viewing Subscriber Sessions in the SSR Database by IP Address (SRC CLI)

Purpose View subscriber sessions and associated service sessions stored in the SSR database by IP address.



NOTE: This command must be executed on a client node.

Enter the start address of the IP range. If you do not specify the **end-address**, only the session with this IP address is displayed.

(Optional) Enter the end address of an IP range. If not specified, only the session that matches **start-address** is displayed.

(Optional) Enter the VPN ID that the sessions belong to. If not specified, only sessions with public IP addresses are displayed. If value is either " " or ' ', display sessions from public network or any VPN.

Enter the maximum number of sessions to display. By default the maximum number of sessions displayed is 25.

Action `user@host> show database subscriber-sessions by-address start-address start-address end-address end-address vpn-id vpn-id maximum-results maximum-result`

```

UserIPAddress    0.0.0.1
VpnID
SessionStartTime 2000-01-01 00:00:00.0
UserName         abc

```

```

Service session
Subscription Name test1
SessionName      DEFAULT

```

```
Subscriber session
```

```
UserIPAddress    0.0.0.2
VpnID
SessionStartTime 2000-01-02 00:00:00.0
UserName        abc
```

```
Service session
Subscription Name test1
SessionName      DEFAULT
```

```
Subscriber session
UserIPAddress    0.0.0.3
VpnID
SessionStartTime 2000-01-03 00:00:00.0
UserName        abc
```

```
Subscriber session
UserIPAddress    0.0.0.4
VpnID
SessionStartTime 2000-01-04 00:00:00.0
UserName        abc
```

- Related Topics**
- Viewing Subscriber Sessions in the SSR Database by Indexed Field (SRC CLI) on page 389
 - Viewing All Subscriber Sessions in the SSR Database (SRC CLI) on page 387
 - Viewing the Total Number of Subscriber Sessions in the SSR Database (SRC CLI) on page 390

Viewing Subscriber Sessions in the SSR Database by Indexed Field (SRC CLI)

Purpose View subscriber sessions and associated service sessions stored in the SSR database based on an indexed field in the subscriber sessions table.

Enter the name of the indexed field in the subscriber sessions table.

Enter the value for the indexed field. For fields of integer or binary type, sessions that match the selected indexed field specified value are displayed. For fields of string type, the wildcard '*' can be used as the value. "*" matches any number of characters. However, the value cannot start with a wildcard, such as "*", or "*abc".

Enter the maximum number of sessions to display. By default the maximum number of sessions displayed is 25.



NOTE: This command can be executed from any client node.

Action `user@host> show database subscriber-sessions by-indexed-field name indexed-field-name value value maximum-results maximum-result`

```
UserIPAddress    0.0.0.1
VpnID
SessionStartTime 2000-01-01 00:00:00.0
UserName        abc
```

```
Service session
```

```
Subscription Name test1
SessionName        DEFAULT
```

```
Subscriber session
UserIPAddress      0.0.0.2
VpnID
SessionStartTime  2000-01-02 00:00:00.0
UserName          abc
```

```
Service session
Subscription Name test1
SessionName        DEFAULT
```

```
Subscriber session
UserIPAddress      0.0.0.3
VpnID
SessionStartTime  2000-01-03 00:00:00.0
UserName          abc
```

```
Subscriber session
UserIPAddress      0.0.0.4
VpnID
SessionStartTime  2000-01-04 00:00:00.0
UserName          abc
```

- Related Topics**
- Viewing Subscriber Sessions in the SSR Database by IP Address (SRC CLI) on page 388
 - Viewing All Subscriber Sessions in the SSR Database (SRC CLI) on page 387
 - Viewing the Total Number of Subscriber Sessions in the SSR Database (SRC CLI) on page 390

Viewing the Total Number of Subscriber Sessions in the SSR Database (SRC CLI)

Purpose View the total number of subscriber sessions currently in the SSR database.



NOTE: This command must be executed on a client node.

Action user@host> show database subscriber-sessions count

```
root@igor> show database subscriber-sessions count
Number of Subscriber Sessions: 4
```

- Related Topics**
- Viewing Subscriber Sessions in the SSR Database by IP Address (SRC CLI) on page 388
 - Viewing Subscriber Sessions in the SSR Database by Indexed Field (SRC CLI) on page 389
 - Viewing All Subscriber Sessions in the SSR Database (SRC CLI) on page 387

PART 8

Using the Subscriber Information Collector

- Overview of the Subscriber Information Collector on page 393
- Configuring the Subscriber Information Collector with the SRC CLI on page 415
- Monitoring the Subscriber Information Collector with the SRC CLI on page 461

Overview of the Subscriber Information Collector

- Subscriber Information Collector Overview on page 393
- Overview of Local and Shared Configurations for the SIC (SRC CLI) on page 394
- Overview of SIC Accounting Methods and Targets (SRC CLI) on page 396
- Overview of SIC Request Routing Rules (SRC CLI) on page 399
- SIC Editing Rules Overview (SRC CLI) on page 401
- Overview of RADIUS Configuration for the SIC (SRC CLI) on page 404
- Overview of RADIUS Transports for the SIC Group and Server on page 407
- Overview of SIC Dictionaries and Device Models (SRC CLI) on page 407
- Overview of Loading SIC Dictionaries (SRC CLI) on page 408
- Overview of SIC Local Realms on page 409
- Overview of SIC Event Logging (SRC CLI) on page 410
- Overview of SNMP Support for the SIC (SRC CLI) on page 413

Subscriber Information Collector Overview

The subscriber information collector (SIC) is used in conjunction with the MX Series Ethernet Services Router running the packet-triggered subscribers and policy control (PTSP) solution. The SIC listens for RADIUS accounting events from IP edge devices (accounting clients), either directly or indirectly through an authentication, authorization, and accounting (AAA) proxy server, allowing the SRC software to gain increased subscriber awareness.

The role of the SIC is to listen for RADIUS accounting events and filter undesired events based on attachment session attributes. The SIC is also responsible for sending RADIUS accounting events to the correct SRC that is managing the MX Series router.

The major components of the SIC are:

- Accounting listeners, which are configured with port numbers and parameters controlling receipt of UDP packets.
- A collection of RADIUS dictionaries.

- A collection of network access server (NAS) clients.
- A collection of RADIUS accounting targets.
- A collection of routing rules.
- A collection of editing rules.
- A collection of RADIUS network elements. A RADIUS network element contains an ordered list of RADIUS accounting clients, targets or both, along with a failover policy for targets.
- A collection of accounting methods including storing accounting events in the SRC session state registrar (SSR) database or forwarding them to a downstream AAA server (network element).
- Components supporting SNMP, statistics, and event logging.

The sequence the SIC uses to process an accounting event is:

1. A RADIUS accounting request is received by a SIC accounting listener.
2. The NAS client is identified by means of the source IP address of the incoming packet.
3. The request is parsed by means of the dictionary associated with the NAS Client.
4. A routing decision is made by means of the configured routing rules.
5. Before the SIC submits the accounting request to the accounting route target, the request is optionally edited by means of the editing rules associated with the selected accounting route rule.
6. The SIC routes the request to the accounting target (either SSR database or target AAA server).

- Related Topics**
- Overview of SIC Accounting Methods and Targets (SRC CLI) on page 396
 - Overview of Local and Shared Configurations for the SIC (SRC CLI) on page 394
 - Overview of SIC Request Routing Rules (SRC CLI) on page 399

Overview of Local and Shared Configurations for the SIC (SRC CLI)

For the SIC you need to define both a local and a shared group configuration.

Local Configuration

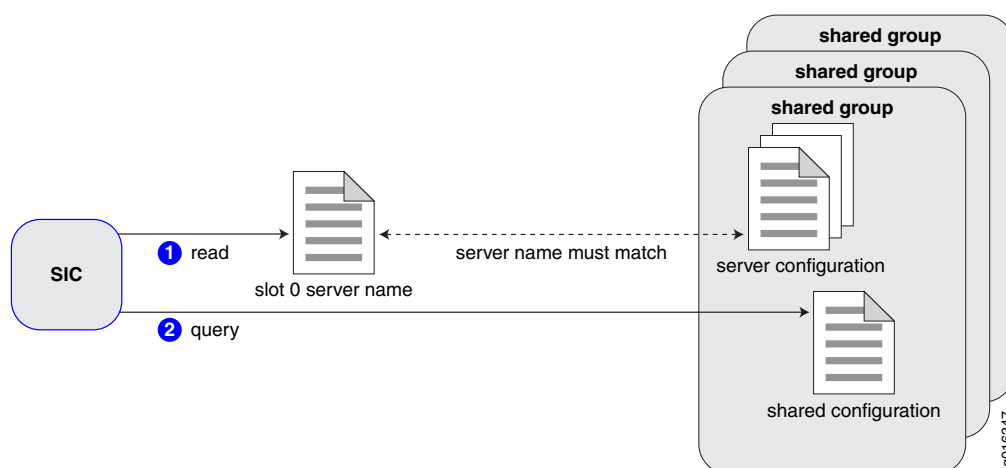
A local configuration applies to a specific server instance in the SIC group. The local configuration specifies the name of the server and the properties the server uses to connect to the Juniper Networks database where the configuration is stored. You specify the local server name by using the **edit slot *number* sic server** statement. You specify the connection properties for the Juniper Networks database by using the **slot *number* sic initial directory-connection** statement.

Shared Configuration

The SIC shared group configuration contains the configuration used by a group of servers. Each SIC server must belong to a group. The SIC group configuration controls the properties for accounting methods, dictionaries, editing rules, and RADIUS options.

When a SIC server starts, it reads its server name from its local configuration. It then queries the Juniper Networks database using that name. The query returns the relevant shared group that contains that server name. Figure 61 on page 395 depicts the process used by the SIC server on startup to access its shared group configuration.

Figure 61: Configuration of SIC Servers and Groups



You configure the SIC shared group with the **shared sic group identifier** statements. The identifier associated with the group is the name of the shared configuration. If you want a specific server to use a shared group configuration, you create a correlation between the server instance and the shared group configuration. Specify the name of the specific server by using the **edit shared sic group identifier server** statement. The identifier associated with the server must match the name that you specified in the **slot number sic server** statement.

In addition, certain configuration options applicable to the individual server instances belonging to the group are also stored in the shared group configuration under the individual server name. These include the accounting route rules, the event logging configuration, and the inbound and outbound RADIUS transport configuration specific to the server instance. You configure these options by using the **edit shared sic group identifier server** statements.

For example, if you want the server named `server-bldg5` to be part of the SIC group named `server-group1`, from configuration mode:

1. Specify the name of the server instance.


```
[edit]
user@host# edit slot 0 sic server
set name server-bldg5
```

2. Specify the SIC group name.

```
[edit]
user@host# edit shared sic group server-group1
```

3. Specify the name of the server instance you want to include in the SIC group configuration.

```
[edit]
user@host# edit shared sic group server-group1 server server-bldg5
```

- Related Topics**
- Creating a SIC Server Instance (SRC CLI) on page 419
 - Creating a SIC Group (SRC CLI) on page 417
 - Adding a Server to a SIC Group (SRC CLI) on page 419

Overview of SIC Accounting Methods and Targets (SRC CLI)

The available types of *accounting methods* for the subscriber information collector (SIC) include:

- Database—Stores accounting events in the SSR database
- Proxy—Forwards accounting events to a downstream network element that contains a proxy AAA server

You configure *accounting targets* by specifying the accounting method used by the SIC group. The accounting target for explicit accounting routing rules can be either the SSR database, or a AAA server in a downstream network element. The accounting target for implicit routing rules is always a proxy AAA server in a remote network element.

Using the SSR Database as the Accounting Method

You use the **shared sic group *identifier* accounting-method *accounting-method-name* database** statement to configure the SSR database as the accounting method. Configure this accounting method as the accounting target by using the **shared sic group *identifier* server *identifier* accounting-route *id* target** statement. In addition, you need to define the mapping between any request attributes, literals, or SIC variables, and the respective SAE plug-in attributes by using the **shared sic group *identifier* accounting-method *accounting-method-name* database plug-in-attribute** statement. You also need to configure the mapping between the SAE plug-in attributes and the columns in the SSR database by using the **shared database cluster (primary) attribute-associations entity *name* field** statement.

Following is a basic working configuration for the database accounting method that includes the default configuration:

- Accounting listener—You must specify a name for the accounting listener. You can use the default port 1813.
- Device model—You can use the default model.

- Database accounting method—This is the default accounting method.
- Upstream RADIUS network element and accounting client—You must define the upstream network element and at least one accounting client.
- Accounting route—You must define the accounting route.
- Logger—You can use the default logger.

Mapping Attributes When Using the Database Accounting Method

When you use the SSR database as the accounting method, you need to define the mapping between:

- SIC request attributes and variables and SAE plug-in attributes
- SAE plug-in attributes and fields in the subscriber sessions table in the SSR database

The SIC uses internal variables to store intermediate results of transaction processing, such as editing. Every variable must have a name and a value. You can define variables while configuring editing rules. If a variable is configured in an editing rule and it does not already exist, it is created. Another place where variables are used is in the mapping between the SIC, SAE plug-in attributes, and the fields in the subscriber sessions table in the SSR database. A variable from an editing rule can be used in the mapping, which allows you to store the value of the variable (the result of the editing process) in the subscriber sessions table field. There are some internal SIC variables such as:

- ReceiveTime—This is the timestamp of the accounting event.
- UserStatusType—This is correlated to the RADIUS Acct-Status-Type: 1 for Accounting-Start, 2 for Accounting-Stop.



NOTE: You must configure the SIC group to use the database accounting method. In addition, you must map any fields in the SSR subscriber sessions table that have a not-null requirement to either a request attribute or SIC variable.

This output shows the attribute mapping between SAE plug-in attributes and the SIC request attributes and variables

```
plug-in-attribute id="PA_USER_INET_ADDRESS" request-attribute="NAS-IP-Address"
plug-in-attribute id="PA_LOGIN_NAME" request-attribute="User-Name"
plug-in-attribute id="PA_PROPERTY.session-start-time" variable="ReceiveTime"
```

This output shows the mapping between SAE plug-in attributes and the fields in subscriber sessions table in SSR database:

```
ssrMapping
table name="SubscriberSessions"
attributeMapping attribute="PA_USER_INET_ADDRESS" field="UserIPAddress"
attributeMapping attribute="PA_LOGIN_NAME" field="UserName"
  attributeMapping attribute="PA_PROPERTY.session-start-time"
  field="SessionStartTime"
    table
      ssrMapping
```

This mapping results in attributes and variables mapped as shown in Table 24 on page 398.

Table 24: Example of SSR Database Mapping

SIC Variable or Attribute	SAE Plug-In Attribute	Field in Subscriber Sessions Table
Request-attribute=NAS-IP-Address	PA_USER_INET_ADDRESS	UserIPAddress
Request-attribute=User-Name	PA_LOGIN_NAME	UserName
Variable=ReceiveTime	PA_PROPERTY.session-start-time	SessionStartTime

Using the Proxy RADIUS Accounting Method

The proxy RADIUS accounting method forwards accounting events to an accounting target (AAA server) located in a downstream network element. You use the **shared sic group identifier accounting-method accounting-method-name proxy radius** statement to configure proxy RADIUS as the accounting method. You configure the downstream network element that contains the AAA server by using the **shared sic group identifier radius network-element id downstream** statement. You configure the AAA server as the accounting target by using the **shared sic group identifier radius network-element id downstream accounting-target** statement.

Following is a basic working configuration for the proxy accounting method that includes the default configuration:

- Accounting listener—You must specify a name for the accounting listener. You can use the default port 1813.
- Device model—You can use the default model.
- Proxy accounting method—You must configure the proxy accounting method.
- Outbound transport—You can use the default outbound transport.
- Upstream RADIUS network element and accounting client—You must define the upstream network element and at least one accounting client.
- Downstream RADIUS network element and accounting target—You must define the downstream network element and the accounting target.
- Accounting route—You must define the accounting route.
- Logger—You can use the default logger.

Related Topics

- Configuring the SSR Database as the Accounting Method (SRC CLI) on page 420
- Configuring Proxy RADIUS as the Accounting Method (SRC CLI) on page 421
- Overview of SIC Request Routing Rules (SRC CLI) on page 399
- Configuring Downstream RADIUS Network Elements and Accounting Targets for the SIC Group (SRC CLI) on page 431
- Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI) on page 439

Overview of SIC Request Routing Rules (SRC CLI)

SIC routing rules define how the SIC routes accounting requests. You configure routing rules for each server in the SIC group. There are two types of accounting routing rules:

- Explicit routing rules
- Implicit routing rules

Explicit Routing Rules

Explicit routing rules consist of a condition, or set of conditions, and an accounting target to which the accounting request is to be routed. Routing criteria consist of a list of simple Boolean expressions on RADIUS attributes and transactional variables. The accounting target is either the SSR database or a downstream network element that contains a AAA server.

You can specify explicit routing rules based on these match conditions:

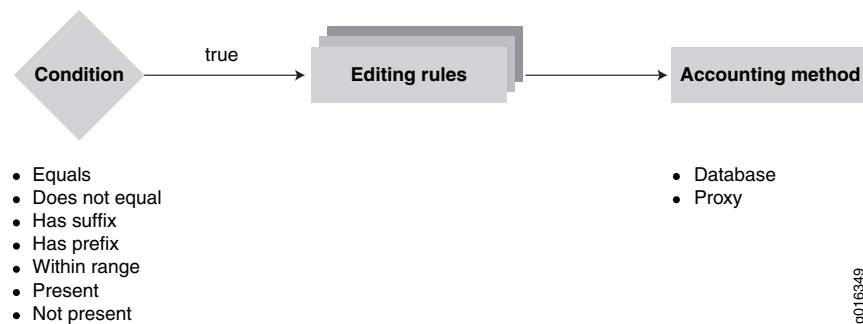
- Realm name
- User identity
- Request attribute

You can test the value of the match condition for the following conditions:

- Present
- Not present
- Equal
- Does not equal
- Has prefix
- Has suffix
- Range

The has prefix and has suffix condition tests work only on string representation of the value. To test for a range condition, specify a low and high value. Figure 62 on page 400 depicts the explicit routing rule process.

Figure 62: Explicit Routing Rule Process



When the SIC receives an accounting request, it evaluates any defined explicit routing rules in the order they were configured. For a route to be selected, *all* conditions of the rule must be true. If a match is not found, the next configured rule is examined, and so on. As soon as a rule with matching criteria is encountered, the iteration stops and the accounting target in that rule is selected as the destination for the request. If a match for all conditions cannot be found in the explicit routing rules, the implicit routing rules are examined.

Before the request is sent to the specified accounting target, you can edit it by optionally specifying an editing rule for the accounting route.

Examples of explicit routing rules are:

Rule 1: If the NAS-Identifier is nas1, then the target is accounting method method1, which is the database accounting method:

```
[edit shared sic group group1 server server1 accounting-route route1]
user@host# show
target method1;
condition {
  attribute nas-identifier;
  equals nas1;
}
```

Rule 2: If the NAS-Identifier is nas2 then the target is accounting method method2, which is the proxy accounting method, pointing to the RADIUS network element rne1:

```
[edit shared sic group group1 accounting-method method2 proxy]
user@host# show
radius {
  network-element rne1;
}
...
[edit shared sic group group1 server server1 accounting-route route2]
user@host# show
target method2;
condition {
  attribute nas-identifier;
  equals nas2;
}
```


Implicit Routing Rules

Implicit routes are realm-based. You configure implicit routes by defining a network element that contains a remote AAA server, and assigning it the proxy function. You can then either define a default route used for all requests from all realms, or you can specify that only requests from specific realms are routed to the proxy AAA server. When you specify realms, you have the option to set a condition of either an exact match of the realm string, or a match on the prefix of the realm string.

Implicit routing rules have lower priority and are evaluated only if a match is not found for explicit routing rules. When a request is received, the SIC server evaluates the associated routing rules. First, the server evaluates any explicit routing rules. If no match is found, the server evaluates the implicit routing rules. When a match is found, the server processes the request by routing it to the specified network element that has the proxy function assigned to it.

- Related Topics**
- Configuring Explicit Accounting Routes for the SIC (SRC CLI) on page 442
 - Configuring Implicit Accounting Routes for the SIC (SRC CLI) on page 444
 - Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI) on page 439
 - SIC Editing Rules Overview (SRC CLI) on page 401
 - Overview of SIC Accounting Methods and Targets (SRC CLI) on page 396

SIC Editing Rules Overview (SRC CLI)

Before the SIC sends the request to the specified accounting target, the request can optionally be edited according to the editing rules associated with the selected routing rule. Editing rules are similar to routing rules, in that the request is examined for matching conditions, and if one is found, the request is edited and then sent to the accounting target.

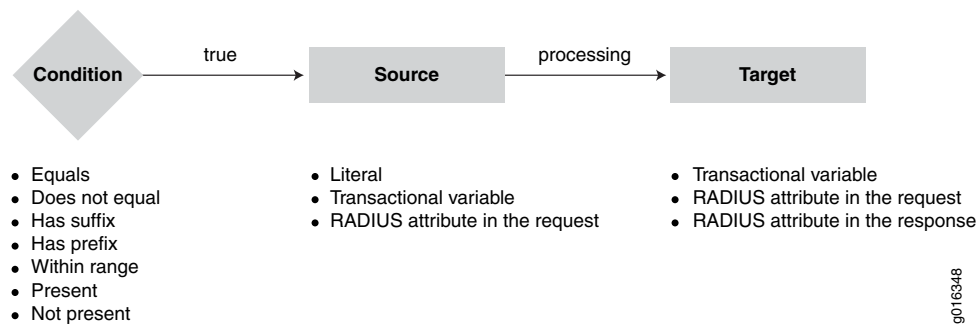
In addition to editing RADIUS attributes, the SIC can edit transactional variables and literals. Editing rules can define new transactional variables, in addition to certain built-in variables, such as the result of username parsing, network access server (NAS) client lookup, and so on. Transactional variables are also referenced in the columns of the subscriber sessions table in the SSR database, thus allowing you to store the results of request processing and editing in the subscriber sessions table.



NOTE: You can control the number of transactional variables the SIC supports. The default value is 255. When you change this limit, you need to restart the SIC.

Figure 63 on page 402 depicts the SIC editing rule process.

Figure 63: SIC Editing Rule Process



You configure editing rules by defining the source and its associated match conditions, the editing conditions applied to the source value, and the target in which the edited result is placed. First the SIC examines the specified source in the request for the defined match conditions. If all conditions are found to be true, the SIC edits the source value based on the defined editing conditions. The result is then placed in the defined target. The edited request, including both the original source and the new target value, is sent to the accounting target.

Each editing rule is a simple assignment of a source (RValue) and a target (LValue). In any assignment the target can be either:

- Transactional variable
- RADIUS attribute in the request
- RADIUS attribute in the response

The source can be either:

- Literal
- Transactional variable
- RADIUS attribute in the request

The match conditions that you can test for in the source include whether a specific realm, user identity, or request attribute is:

- Present
- Not present
- Equals
- Does not equal
- Has suffix
- Has prefix
- Within range

If a match condition is found on the source, you can append or replace the value of the source and place it in the target. Additionally, if the source is a request attribute, you can

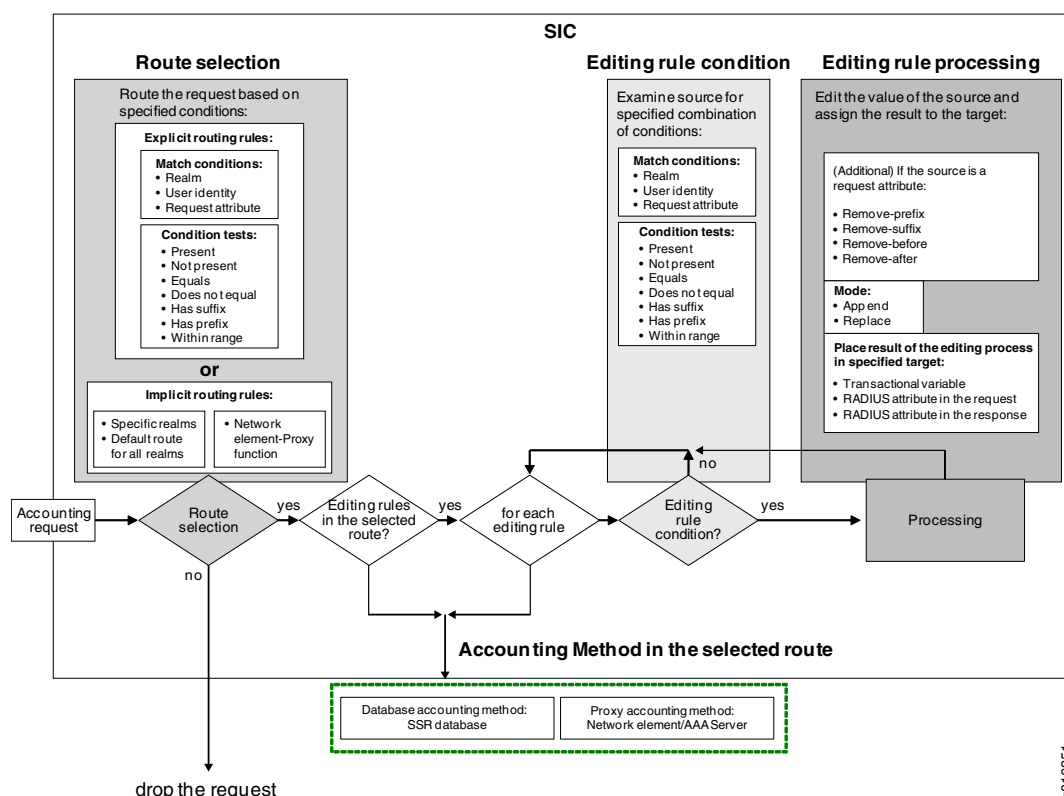
edit the value of the source by removing the suffix or prefix, or removing what is before or after the @ and place the result in the target. The remove before @ and remove after @ options can contain wildcards.

For example, if the request contains johnsmith@abcd.net:

- Removing the prefix john results in: smith@abcd.net
- Removing the suffix .net results in: johnsmith@abcd
- Remove the attribute before the @ results in: abcd.net
- Remove the attribute after the @ results in: johnsmith

Figure 64 on page 403 depicts the editing rule process and the accounting route selection process.

Figure 64: Editing and Accounting Routing Rule Conditions and Processes



Example of an editing rule:

1. If Unisphere-Virtual-Router is present, then the transactional variable vpn-id is the substring after ':' in Unisphere-Virtual-Router.
2. If the NAS-Identifier is nas3, then the transactional variable vpn-id is the realm portion of User-Name (realm transactional variable).
3. Otherwise the transactional variable vpn-id is the NAS-Identifier.

```
[edit shared sic group group1 editing edit1]
user@host# show
target {
  variable vpn-id;
}
source {
  request-attribute Unisphere-Virtual-Router {
    condition {
      attribute Unisphere-Virtual-Router;
      check-presence;
    }
    remove-before *:
  }
  variable realm {
    condition {
      attribute nas-identifier;
      equals nas1;
    }
  }
}
default {
  request-attribute nas-identifier ;
}
```

- Related Topics**
- [Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\) on page 439](#)
 - [Configuring Explicit Accounting Routes for the SIC \(SRC CLI\) on page 442](#)
 - [Overview of SIC Accounting Methods and Targets \(SRC CLI\) on page 396](#)
 - [Configuring the SSR Database as the Accounting Method \(SRC CLI\) on page 420](#)
 - [Configuring Proxy RADIUS as the Accounting Method \(SRC CLI\) on page 421](#)

Overview of RADIUS Configuration for the SIC (SRC CLI)

The RADIUS configuration for the SIC group consists of:

- Accounting listeners for the SIC group
- Outbound transport for the SIC group
- At least one upstream network element and accounting client
- (Optional) A downstream network element and accounting target
- (Optional) A proxy function—Used for defining implicit routing rules

Accounting Listener

The accounting listener listens for RADIUS accounting events and filters undesired events based on attachment session attributes. You must configure at least one accounting listener for the SIC group. To configure the accounting listener, you specify the UDP port that the SIC listens on, as well as other parameters that control the receipt of UDP packets. The configuration options associated with the accounting listener control the RADIUS inbound transport for the SIC group, which is used to communicate with upstream network elements that contain one or more accounting clients.

Outbound Transport for the SIC Group

The RADIUS outbound transport for the SIC group controls communication to accounting targets that reside in downstream network elements. You need to configure the outbound transport for the SIC group only if you are using the proxy accounting method or if you are using the proxy function for implicit routing. You do not need to configure the outbound transport when you are using the database accounting method. You configure the RADIUS outbound transport that the SIC group sends messages on by specifying the UDP port and connect and disconnect related configuration options.

Inbound and Outbound Transport for the SIC Server

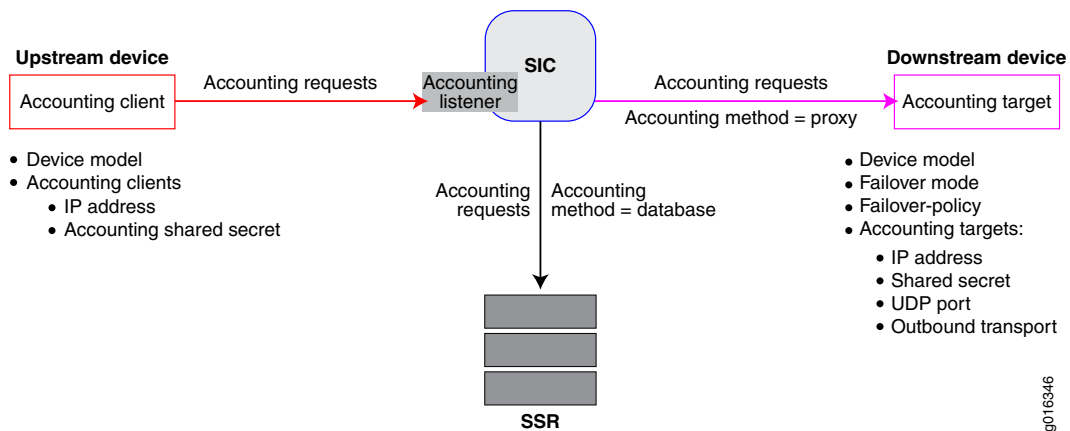
SIC servers use the same inbound and outbound transports that are configured for the SIC group. You configure the inbound transport for the SIC server by specifying the name of the transport for the accounting listener. You configure the outbound transport for the SIC server by specifying the name of the outbound transport configured for the SIC group.

Network Elements

A network element is an addressable, logical network entity. An upstream network element contains accounting clients, which send accounting messages to the SIC accounting listener. A downstream network element contains a AAA server to which the SIC forwards, or proxies, accounting messages when the SIC is configured for the proxy accounting method. An upstream network element can contain multiple accounting clients, and a downstream network element can contain multiple accounting targets.

Figure 65 on page 405

Figure 65: Upstream and Downstream Network Elements



Overview of Configuring Upstream Network Elements

You need to configure at least one upstream network element and accounting client. You configure upstream network elements by using the **shared sic group group-name radius network-element upstream** statement, and specifying the shared secret, IP address, and device model of the accounting client.

Overview of Configuring Downstream Network Elements

You need to configure a downstream network element only if you are using the proxy accounting method. A downstream network element contains an accounting target (AAA server) to which the SIC forwards accounting requests. You configure downstream network elements by using the **shared sic group *group-name* radius network-element downstream** statement, and specifying the outbound transport, UDP port, shared secret and IP address of the accounting target. You also need to specify the failover policy, failover mode and the device model of the accounting target (AAA server).

Failover Policy

The failover policy controls how the SIC server responds when it does not receive a response from an accounting target within a specified amount of time. When the SIC server sends a message to an accounting target, it expects to receive a reply within a certain amount of time as specified by the fast fail policy. If it does not receive the reply in the specified time, it places the accounting target into fast fail mode and rejects the request. You configure the failover policy by specifying the fast fail and retry parameters, which control such options as minimum number of times the server retransmits messages to an accounting target, delay between sending retransmissions, as well as various time out and delay settings that control how fast the server goes in and out of fast fail mode.

Failover Mode

The failover mode manages how the SIC server sends messages over multiple paths to a downstream network element (accounting target). You can configure the failover mode for either round robin or primary/backup. When the server has a message to send to a downstream network element, it first examines whether the failover mode is set for round robin or primary/backup. Next, it examines whether all paths to the network element are operational. It then sends the message over whatever path is operational based on the failover mode. When you use the round robin method, the server alternates the path it uses to the network element. When you use the primary/backup method, the server sends all messages over the first path defined in an ordered list. If the first path fails, all messages are sent over the next path in the ordered list. When the first path becomes operational again, all messages are again sent over it.

Using the Proxy Function to Define Implicit Routing Rules

You use the proxy function to define implicit routing rules by specifying a remote AAA server as a proxy, and having the SIC forward accounting requests to it. When the SIC receives a request, it first evaluates any explicit routing rules. If no match is found, it evaluates implicit routing rules. If a match is found, the SIC routes the request to the proxy AAA server.

You configure the proxy function by configuring a network element and specifying it as a proxy. You can then either define a default route used for all requests from all realms, or you can specify that only requests from certain realms are routed to the proxy AAA server. When you specify realms, you have the option to specify a match condition of either an exact match of the realm string, or a match on the prefix of the realm string.

- Related Topics**
- Configuring the RADIUS Accounting Listener for the SIC Group (SRC CLI) on page 422
 - Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI) on page 423

- [Configuring Implicit Accounting Routes for the SIC \(SRC CLI\) on page 444](#)
- [Overview of SIC Request Routing Rules \(SRC CLI\) on page 399](#)
- [Overview of RADIUS Transports for the SIC Group and Server on page 407](#)

Overview of RADIUS Transports for the SIC Group and Server

You need to configure the RADIUS transports for the SIC group and for each server in the group.

You configure the inbound transport for the SIC group by configuring the accounting listener. You need to configure the outbound transport if you are using the proxy accounting method, or if you are using implicit routing to forward accounting messages to a remote AAA server acting as a proxy. You do not need to configure the outbound transport if you are using the database accounting method.

The servers contained in the SIC group use the same inbound and outbound transport names as configured for the SIC group. In addition, on the inbound side of the server you need to specify the IP address used by the server for receiving UDP packets.

- Related Topics**
- [Overview of RADIUS Configuration for the SIC \(SRC CLI\) on page 404](#)
 - [Configuring the RADIUS Accounting Listener for the SIC Group \(SRC CLI\) on page 422](#)
 - [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 423](#)
 - [Configuring the RADIUS Transport for a SIC Server \(SRC CLI\) on page 425](#)

Overview of SIC Dictionaries and Device Models (SRC CLI)

The SIC uses dictionaries to define RADIUS attributes. Dictionaries identify the attributes the SIC expects when receiving RADIUS requests from a specific type of device and the attributes the SIC includes when sending a RADIUS response to a specific type of device. The SIC uses these definitions to parse accounting requests and generate responses.

Dictionaries and the Device Models Supported by the SIC Group

Each SIC group configuration must include a dictionary and a list of device models supported by the group. When you configure the device model, you specify an identifier and the associated dictionary that the SIC uses when communicating with the device. The dictionary assigned to the device model identifies the attributes the SIC expects when receiving RADIUS requests from the specific device, and the attributes the SIC needs to include in responses to the device. The SIC uses these definitions to parse accounting requests and generate responses.

In addition, when you configure an upstream or downstream network element, you need to specify its device model based on the list of device models you have configured for the SIC group. Thereafter, whenever the SIC receives a RADIUS packet from the network element, it consults the associated dictionary for the attributes that it encounters in the packet.

Overview of Configuring Device Models and Their Associated Dictionaries for the SIC Group

You need to specify the devices models and their associated dictionaries for the SIC group and each network element the SIC needs to communicate with. To specify these for the SIC group, use the **shared sic group identifier model id** configuration statement, and to specify these for network elements, use the **shared sic group identifier radius network-element id upstream** and **shared sic group identifier radius network-element id downstream** configuration statements.

Modifying a Dictionary

You can add attributes or modify existing attributes in dictionaries. However, you cannot delete the dictionary itself or any of the existing attributes. If you modify a dictionary, you need to restart the SIC for the change to take effect. Use the **shared sic group group name dictionary id** configuration statement to modify an existing dictionary.

Overview of Configuring the Dictionaries Used By the SIC Group

The SIC comes with a default dictionary called radius, which is preloaded in the default SIC group called default-group. This dictionary contains attributes defined by the RADIUS standard and is sufficient for most environments. The SIC will not work without a dictionary; every shared SIC group configuration must contain a dictionary. You can include multiple dictionaries in the SIC shared group configuration.

If you create a new SIC group, you must load a dictionary into the shared group configuration. We recommend that you use the default RADIUS dictionary. See “Loading a Dictionary into the SIC Shared Group Configuration (SRC CLI)” on page 428 for complete details on loading dictionaries into the SIC shared configuration.

- Related Topics**
- Configuring the Device Models Supported by the SIC Group (SRC CLI) on page 429
 - Configuring Dictionaries for the SIC Group (SRC CLI) on page 426
 - Loading a Dictionary into the SIC Shared Group Configuration (SRC CLI) on page 428
 - Overview of Loading SIC Dictionaries (SRC CLI) on page 408

Overview of Loading SIC Dictionaries (SRC CLI)

If you create a new SIC group, you need to load a dictionary into the shared group configuration. We recommend that you use the default RADIUS dictionary. You load this dictionary into the new shared SIC group configuration by navigating to the dictionary using the **shared sic group group-name dictionary** statement, and then executing the **load radius dictionary** command. For example, to load the default dictionary into a new SIC group called g2:

```
[edit shared sic group g2 dictionary]
user@host# load radius dictionary radius
Load Dictionary Complete
```

The **load radius dictionary** command can also be used to create a new dictionary. If the shared SIC group configuration already includes a dictionary of the same name you are

loading, you need to specify how you want the dictionaries merged by setting the merge option. If you set the merge option to *replace*, the dictionary object in the Juniper Networks database is replaced with the external dictionary file. As a result, all prior configuration under the existing dictionary object in the Juniper Networks database is lost, and all attributes in the external dictionary file are placed under the dictionary object in the Juniper Networks database. Setting the merge option to *merge*, merges the dictionary object in the Juniper Networks database with the external dictionary file. As a result, all attributes in the external dictionary file are added under the dictionary object in the Juniper Networks database. If there is a conflicting statement, the statement in the external dictionary file is used. If a dictionary with the same name already exists and you do not specify the merge option, then the command ends with an error and the configuration is not modified.



CAUTION: Use caution when loading dictionaries into the shared SIC group configuration. You can merge the dictionaries or replace the existing dictionary. If you choose to replace the dictionary, be aware that all contents in the existing dictionary are replaced by the new dictionary.

It is necessary to load a dictionary only if you create a new shared SIC group configuration. The SIC will not work without a dictionary. If the SIC group already has a dictionary loaded and you want to add or modify an attribute, edit the dictionary using the `shared sic group identifier dictionary` statement.



NOTE: Loading the proper dictionary is one of the most important steps when you configure the SIC shared group configuration. The SIC requires the basic RADIUS attributes such as State, Class, User-Name, and so on, to properly process accounting messages. If you choose to create a new SIC group, we recommend that you load the default dictionary called `radius` into the shared group configuration. You can modify this dictionary by adding new attributes or modifying existing attributes. To create a new dictionary, we recommend that you work with Juniper Networks Technical Support.

Related Topics

- Overview of SIC Dictionaries and Device Models (SRC CLI) on page 407
- Configuring the Device Models Supported by the SIC Group (SRC CLI) on page 429
- Configuring Dictionaries for the SIC Group (SRC CLI) on page 426
- Loading a Dictionary into the SIC Shared Group Configuration (SRC CLI) on page 428

Overview of SIC Local Realms

Defining a realm as local to the SIC group instructs the SIC to use a local server to process the request. The network access identifier (NAI) in the request identifies the intended realm. To properly interpret requests received from intermediate servers, the SIC server must know which realms it is responsible for servicing locally.

When a request is received, the SIC examines the NAI to determine the realm to which the request is to be routed. If the realm is configured as a local realm to the SIC server, the request is processed by the local server. If no realm is present in the NAI, the request is considered to be local.

- Related Topics**
- Configuring What Realms Are Local to the SIC Group (SRC CLI) on page 435
 - Overview of Local and Shared Configurations for the SIC (SRC CLI) on page 394
 - Creating a SIC Server Instance (SRC CLI) on page 419
 - Adding a Server to a SIC Group (SRC CLI) on page 419

Overview of SIC Event Logging (SRC CLI)

You use the SIC log streams to capture different groups of server-related events at various levels of granularity. You can configure the SIC to capture any number of log streams. If you configure multiple log streams, make sure you configure unique names for each log stream by using the **shared sic group identifier server identifier logger identifier** statement. Each log stream you create captures events in a separate log file, which is date stamped, and you can also assign a prefix to it for easy identification.

Log messages are divided into several log groups according to the subject of the log information. You can configure a log stream to display only log messages from particular log groups. Each log group captures different types of server-related events. You configure the level of granularity captured for the log group by setting the event level for the log group.

Log File Options

You use the configuration options described in Table 25 on page 410 to define the properties of the log files.

Table 25: SIC Log File Options

Option	Description
filename	Prefix added to the log file name. This string will be prepended to each log file name.
filter	<p>Filter to define which event messages are logged or ignored. The filter specifies the logging level, such as debug.</p> <ul style="list-style-type: none"> • Error events are captured for every log group • Debug events are captured for every log group
flush-after-writes	<p>If set to true, log messages are immediately written to the log file without buffering. Use this setting for real-time logging.</p> <p>If set to false, SIC log messages are kept in the buffer until the buffer is full and then all messages in the buffer are written to the log file. Use this setting for performance optimization, when real-time logging is not needed.</p>

Table 25: SIC Log File Options (*continued*)

Option	Description
footer	Footer message added to the end of each log file.
header	Header message added to the beginning of each log file.
high-resolution-timestamps	High-resolution time reporting system functions are used.
maximum-file-size	New log file created after these many bytes. When a log file reaches this size, logging will begin in a new log file.
prepend-message-header	Prepend each log message with additional information. Add time, thread, and transaction information to each log message. You can achieve additional fine tuning by using the work-id-label, work-id-padding, and utc options.
rollover-interval	New log file is created after this amount of time elapses. Specified in seconds.
rollover-on-startup	New log file is created every time the server starts.
utc	Time and date values reflect Universal Time Coordinates (UTC), formerly known as Greenwich Mean Time or (GMT). Otherwise, values reflect local time.
work-id-label	Work data ID prefix added to each log message.
work-id-padding	String added to each log message if work data is not available.

Event Levels

The event level specifies the level of detail captured for the log group. You configure the event level by specifying the log group and then specifying the associated event level. The event level you specify is the highest event level displayed for the log group. You can configure the log stream to display log items from levels at and below a particular event level.

Be careful when using event logging because it consumes server resources while capturing events, and consumes disk space to store the log files. We recommend that event logging be used primarily for troubleshooting purposes, and that you limit the amount of information captured in a log stream to control the consumption of server resources and disk space. Limiting the amount of information in the log stream also makes it easier to interpret the information in the log files. For example, you might configure one log stream to capture only configuration-related events by setting the Configuration log group event level to Detail, and setting all other log group event levels to Error.

In general, each event level includes less verbose event types. For example, if you configure an event level of Warning, then warnings and errors are logged to the specified log stream. The event levels in order of increasing verbosity are shown in Table 26 on page 412.

Table 26: SIC Event Levels

Event Level	Description
None	No events will be logged for the log group.
Error	An error as an event that may cause the system to operate incorrectly. Examples include exceptions being thrown, an inability to continue processing a transaction, or configuration errors that cause a component to fail to start.
Warning	Errors and warnings are logged. Warnings are less severe but more verbose than errors, in that a warning should be logged when the system was able to handle an unexpected input or condition without any threat to the operation of the server. Examples of warnings include invalid packet contents or failures in contacting remote servers.
Standard	Errors, warnings, and standard messages are logged. Standard logging messages show events as a result of normal operation.
Detail	Messages in the log are shown at event levels error, warning, standard, and detail. Detail logging is intended to inform why and how the particular result indicated by standard logging was reached. Server components that perform significant processing on the transaction, such as determining validity of the packet contents, log details about decisions they made. All server components that route the transaction through different processing based on the nature of the transaction log their routing activity at this level. The detail log is allowed to refer to the contents of messages logged at the standard level; that is, it will never be read without the standard messages.
Debug	Messages in the log are shown at event levels error, warning, standard, detail, and debug. Debug logging is provided for engineering troubleshooting only.

Log Groups

Log groups specify the type of server functionality for which you want to log events. The log groups listed in Table 27 on page 412 are available.

Table 27: SIC Log Groups

Log Group	Description
Administration	Reports events related to server administration, such as: <ul style="list-style-type: none"> A server access log, including identity of the administrator. This is available using the standard event level. Changes made to the server configuration, including identity of the administrator. This is available using the Detail event level.
Configuration	Reports events related to configuration.
Packet	Reports events related to transaction processing.
PacketTrace	Displays content of a packet in a <attribute name>:<attribute value> format.
PacketTraceRaw	Displays content of a packet in its raw (octets) format.

Table 27: SIC Log Groups (*continued*)

Log Group	Description
System	<p>Reports events related to the system, such as:</p> <ul style="list-style-type: none"> • Resource failures (no memory, file not found, disk full, etc.) • Unknown exceptions • System start • System stop

- Related Topics**
- Configuring Event Logging for a SIC Server (SRC CLI) on page 445
 - Configuring SNMP for the SIC Group (SRC CLI) on page 448
 - Overview of SNMP Support for the SIC (SRC CLI) on page 413

Overview of SNMP Support for the SIC (SRC CLI)

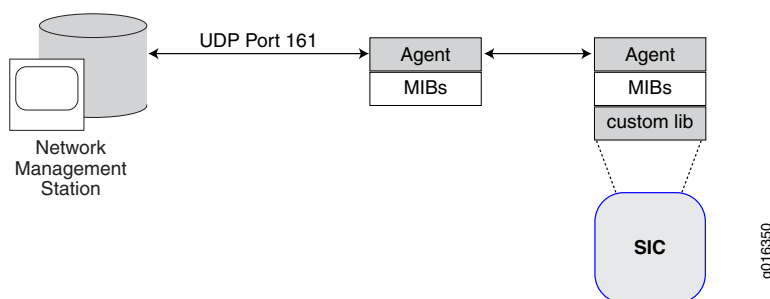
The Simple Network Management Protocol (SNMP) implementation for the SIC supports alerts and is based on an external SNMP agent that accesses the SIC server agent.



NOTE: You can query status using an SNMP Get command, and you can receive alert notifications through SNMP traps from the agent manager. However, you cannot configure the SIC server using SNMP because the SIC does not support the SNMP Set command.

SNMP support for the SIC functions in the same manner as other SRC components. The SIC has its own subagent, which communicates with the main SRC SNMP daemon using the AgentX protocol. The subagent registers with the main daemon for all relevant object identifiers (OIDs). Traps are communicated from the subagent to the main daemon. Figure 66 on page 413 depicts the SIC SNMP support.

Figure 66: SNMP Support for the SIC



Use the **show** command to view all statistics counters available to the SNMP daemon. Table 28 on page 414 list the MIBs used by the SIC to maintain accounting statistics.

Table 28: MIBs Used by the SIC for SNMP Statistics

MIB	Description
RFC4670.mib	Maintains accounting client statistics.
RFC4671.mib	Maintains accounting server statistics.

The SIC supports the traps described in Table 29 on page 414.

Table 29: SNMP Traps Supported for the SIC

SNMP Trap	Description
sic-server-internal-error	A SIC server implementation–dependent error occurred.
sic-server-log-file-failure	Operation on a SIC server log file such as opening, reading, or writing failed. This is most likely due to the server disk being out of space.
sic-server-resource-failure	A SIC server implementation dependent resource failure occurred.
sic-server-shutdown	The SIC server process stopped successfully.
sic-server-startup	The SIC server process started successfully.
sic-server-unauthorized-administration-request	HTTP/HTTPs requests sent to the SIC server were denied because the user does not have proper permission to access the URL.

- Related Topics**
- Configuring SNMP for the SIC Group (SRC CLI) on page 448
 - Configuring Event Logging for a SIC Server (SRC CLI) on page 445
 - Overview of SIC Event Logging (SRC CLI) on page 410

Configuring the Subscriber Information Collector with the SRC CLI

- Configuring the Connection Between the SIC and the Juniper Networks Database (SRC CLI) on page 416
- Creating a SIC Group (SRC CLI) on page 417
- Configuring a Basic SIC Group (SRC CLI) on page 417
- Creating a SIC Server Instance (SRC CLI) on page 419
- Adding a Server to a SIC Group (SRC CLI) on page 419
- Configuring the SSR Database as the Accounting Method (SRC CLI) on page 420
- Configuring Proxy RADIUS as the Accounting Method (SRC CLI) on page 421
- Configuring the RADIUS Accounting Listener for the SIC Group (SRC CLI) on page 422
- Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI) on page 423
- Configuring the RADIUS Transport for a SIC Server (SRC CLI) on page 425
- Configuring Dictionaries for the SIC Group (SRC CLI) on page 426
- Loading a Dictionary into the SIC Shared Group Configuration (SRC CLI) on page 428
- Configuring the Device Models Supported by the SIC Group (SRC CLI) on page 429
- Configuring Upstream RADIUS Network Elements and Clients for the SIC Group (SRC CLI) on page 430
- Configuring Downstream RADIUS Network Elements and Accounting Targets for the SIC Group (SRC CLI) on page 431
- Configuring What Realms Are Local to the SIC Group (SRC CLI) on page 435
- Configuration Statements for SIC Editing Rules (SRC CLI) on page 435
- Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI) on page 439
- Configuration Statements for SIC Explicit Accounting Routing Rules on page 441
- Configuring Explicit Accounting Routes for the SIC (SRC CLI) on page 442
- Configuring Implicit Accounting Routes for the SIC (SRC CLI) on page 444
- Configuring Event Logging for a SIC Server (SRC CLI) on page 445
- Configuring SNMP for the SIC Group (SRC CLI) on page 448
- Example: Basic SIC Group Configuration (SRC CLI) on page 449

Configuring the Connection Between the SIC and the Juniper Networks Database (SRC CLI)

The configuration of the subscriber information collector (SIC) is stored in the Juniper Networks database.

Use the following statements to configure the connection between the SIC and the Juniper Networks database:

```
slot number sic initial directory-connection {  
    url url ;  
    port port ;  
    principal principal ;  
    credentials credentials ;  
    entry-dn entry-dn ;  
    filter filter ;  
}
```

To configure the directory connection properties for the SIC:

1. From configuration mode, access the statement that configures the directory configuration for the SIC in a slot.

```
user@host# edit slot number sic initial directory-connection
```

For example:

```
user@host# edit slot 0 sic initial directory-connection
```

2. Specify the password with which the SIC accesses the directory.

```
[edit slot 0 sic initial directory-connection]  
user@host# set credentials credentials
```

3. (Optional) Specify the URL that identifies the location of the primary directory server.

```
[edit slot 0 sic initial directory-connection]  
user@host# set url url
```

On a C Series Controller, this value is [ldap://127.0.0.1:389](#).

4. (Optional) Specify the port to use when connecting to Juniper Networks database.

```
[edit slot 0 sic initial directory-connection]  
user@host# set port port
```

5. (Optional) Specify the DN that contains the username that the directory server uses to authenticate the SIC.

```
[edit slot 0 sic initial directory-connection]  
user@host# set principal principal
```

6. (Optional) Specify where the root of the SIC configuration is in the directory.

```
[edit slot 0 sic initial directory-connection]
```



```
user@host# set entry-dn entry-dn
```

7. (Optional) Specify any query filters you want to use to monitor changes in the Juniper Networks database.

```
[edit slot 0 sic initial directory-connection]  
user@host# set filter filter
```

8. (Optional) Verify your configuration.

```
[edit slot 0 sic initial directory-connection]  
user@host# show  
url ldap://127.0.0.1:389/  
principal cn=conf,o=Operators,<base>;  
credentials *****;
```

- Related Topics**
- Example: Basic SIC Group Configuration (SRC CLI) on page 449
 - Configuring Initial Directory Eventing Properties for SRC Components
 - Verifying the Local Configuration for a Component

Creating a SIC Group (SRC CLI)

The SIC group configuration controls the properties for accounting methods, dictionaries, editing rules, and RADIUS options.

To create a SIC group:

- From configuration mode, access the configuration statement that creates a SIC group.

```
[edit]  
user@host# edit shared sic group identifier
```

- Related Topics**
- Example: Basic SIC Group Configuration (SRC CLI) on page 449
 - Overview of Local and Shared Configurations for the SIC (SRC CLI) on page 394
 - Adding a Server to a SIC Group (SRC CLI) on page 419

Configuring a Basic SIC Group (SRC CLI)

The SIC default configuration satisfies the needs of most environments, with minor changes such as accounting methods, editing rules, routing rules, and radius clients.

To configure a basic SIC group:

1. Use the default settings for the connection between the SIC and the Juniper Networks database, or configure your own connection.
See “Configuring the Connection Between the SIC and the Juniper Networks Database (SRC CLI)” on page 416.
2. Use the default SIC group (default-group), or create a SIC group.
See “Creating a SIC Group (SRC CLI)” on page 417.
3. Use the default server (default-server), or create a SIC server instance.
See “Creating a SIC Server Instance (SRC CLI)” on page 419.
4. Configure the accounting method used by the SIC group. The session state registrar (SSR) database is the default accounting method.
See “Configuring Proxy RADIUS as the Accounting Method (SRC CLI)” on page 421.
See “Configuring the SSR Database as the Accounting Method (SRC CLI)” on page 420.
5. Configure the accounting listener for the SIC group.
See “Configuring the RADIUS Accounting Listener for the SIC Group (SRC CLI)” on page 422.
6. (Optional) If you are using the proxy accounting method, use the default outbound transport, or configure the outbound transport for the SIC group. This step is also required if you configure a proxy authentication, authorization, and accounting (AAA) server for implicit routing.
See “Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI)” on page 423
7. Configure the transport options for each SIC server in the group.
See “Configuring the RADIUS Transport for a SIC Server (SRC CLI)” on page 425.
8. If you create a new SIC group instead of using the default group, load the default RADIUS dictionary into the shared group configuration.
See “Loading a Dictionary into the SIC Shared Group Configuration (SRC CLI)” on page 428.
9. Use the default device model (default-model), or configure the device models used by upstream and downstream network elements.
See “Configuring the Device Models Supported by the SIC Group (SRC CLI)” on page 429.
10. Configure an upstream RADIUS network element and at least one accounting client.
See “Configuring Upstream RADIUS Network Elements and Clients for the SIC Group (SRC CLI)” on page 430.
11. (Optional) If you are using the proxy accounting method, configure a downstream network element and an accounting target.
See “Configuring Downstream RADIUS Network Elements and Accounting Targets for the SIC Group (SRC CLI)” on page 431.

12. (Optional) Configure any realms you want processed by the local server.
See “Configuring What Realms Are Local to the SIC Group (SRC CLI)” on page 435.
13. (Optional) Configure editing rules for the SIC group.
See “Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI)” on page 439.
14. Configure the accounting routes.
See “Configuring Explicit Accounting Routes for the SIC (SRC CLI)” on page 442.
See “Configuring Implicit Accounting Routes for the SIC (SRC CLI)” on page 444.
15. Configure event logging.
See “Configuring Event Logging for a SIC Server (SRC CLI)” on page 445.

Related Topics • Example: Basic SIC Group Configuration (SRC CLI) on page 449

Creating a SIC Server Instance (SRC CLI)

Use the following configuration statement to configure the SIC server instance:

```
slot number sic server {  
    name name;  
}
```

To create an instance of a SIC server:

- From configuration mode, access the statement that configures the SIC server.

```
[edit]  
user@host# edit slot 0 sic server  
set name name
```

Related Topics • Example: Basic SIC Group Configuration (SRC CLI) on page 449

- Overview of Local and Shared Configurations for the SIC (SRC CLI) on page 394
- Creating a SIC Group (SRC CLI) on page 417
- Adding a Server to a SIC Group (SRC CLI) on page 419
- Configuring Event Logging for a SIC Server (SRC CLI) on page 445

Adding a Server to a SIC Group (SRC CLI)

To add a server to a SIC group:

- From configuration mode, access the statement that configures which servers belong to the SIC group. For example, to add the SIC server called `server1` to the group called `group1`:

[edit]

user@host# edit shared sic group *identifier* server *identifier*

- Related Topics**
- Example: Basic SIC Group Configuration (SRC CLI) on page 449
 - Overview of Local and Shared Configurations for the SIC (SRC CLI) on page 394
 - Creating a SIC Server Instance (SRC CLI) on page 419
 - Configuring Event Logging for a SIC Server (SRC CLI) on page 445
 - Configuring the RADIUS Transport for a SIC Server (SRC CLI) on page 425

Configuring the SSR Database as the Accounting Method (SRC CLI)

When you use the SSR database as the accounting method for the SIC group, accounting events are stored in the SSR database. Part of configuring this accounting method includes configuring the mapping between any request attributes, literals, or SIC variables, and the respective SAE plug-in attributes.



NOTE: At a minimum, you must configure the mapping between the SIC and the SAE plug-in attributes `user-inet-address` and `vpin-id`. You also need to map these SAE plug-in attributes to the primary keys: `UserIPAddress` and `VpnID` in the SSR subscriber sessions table.

Use the following statements to configure the SSR database as the accounting method for the SIC group:

```
shared sic group identifier accounting-method accounting-method-name
```

```
shared sic group identifier accounting-method accounting-method-name database {  
}
```

```
shared sic group identifier accounting-method accounting-method-name database  
  plug-in-attribute id {  
    request-attribute request-attribute;  
    variable variable;  
    literal literal;  
  }
```

To configure the SSR database as the accounting method for the SIC group:

1. From configuration mode, access the statement that configures the SSR database as the accounting method. This sample procedure uses `group1` as the group identifier and `acm1` as the accounting method name.

[edit]

user@host# edit shared sic group `group1` accounting-method `acm1` database

2. Specify the mapping between the SAE plug-in attribute and the request attribute, SIC variable, or literal. For example, to map the SAE plug-in attribute login-name to the request attribute User-Name:

```
[edit shared sic group group1 accounting-method acm1 database ]
user@host# edit plug-in-attribute login-name
user@host# set request-attribute User-Name
```

3. (Optional) Verify your configuration.

```
[edit shared sic group g1 accounting-method acm1 database]
user@host# show
plug-in-attribute {
  login-name {
    request-attribute User-Name;
  }
}
```

4. Configure the mapping between the SAE plug-in attribute and the field in the subscriber sessions table in the SSR database. For information about configuring this mapping, see “Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table (SRC CLI)” on page 371.

- Related Topics**
- Example: Basic SIC Group Configuration (SRC CLI) on page 449
 - Overview of SIC Accounting Methods and Targets (SRC CLI) on page 396
 - Overview of SIC Request Routing Rules (SRC CLI) on page 399

Configuring Proxy RADIUS as the Accounting Method (SRC CLI)

When you use the proxy RADIUS accounting method, the SIC forwards accounting messages to a remote AAA server (accounting target) located in a downstream network element for processing.

To use the proxy RADIUS accounting method, you need to have previously configured the downstream network element, associated accounting target, and committed the configuration. For details about configuring downstream network elements and accounting targets, see “Configuring Downstream RADIUS Network Elements and Accounting Targets for the SIC Group (SRC CLI)” on page 431.

Use the following statements to configure the proxy accounting method:

```
shared sic group identifier accounting-method accounting-method-name
```

```
shared sic group identifier accounting-method accounting-method-name proxy radius {
  network-element network-element;
}
```

To configure proxy RADIUS as the accounting method for the SIC group:

1. From configuration mode, access the statement that configures the accounting method. For example, to configure an accounting method called `acm2` for the SIC group `group2` and specify proxy RADIUS as the accounting method:

```
[edit]
user@host# edit shared sic group group2 accounting-method acm2 proxy radius
```

2. Specify the name of the previously configured downstream network element that contains the AAA server (accounting target) you want the SIC to forward accounting events to. For example, to forward accounting events to the downstream network element `ne2`:

```
[edit shared sic group group2 accounting-method acm2 proxy radius]
user@host# set network-element ne2
```

- Related Topics**
- Example: Basic SIC Group Configuration (SRC CLI) on page 449
 - Overview of SIC Accounting Methods and Targets (SRC CLI) on page 396
 - Overview of SIC Request Routing Rules (SRC CLI) on page 399

Configuring the RADIUS Accounting Listener for the SIC Group (SRC CLI)

The accounting listener listens for RADIUS accounting events and filters undesired events based on attachment session attributes. You need to specify the UDP port and configure the accounting listener options that control the receipt of UDP packets.

Use the following statements to configure the RADIUS accounting listener for the SIC group:

```
shared sic group identifier radius accounting-listener limit {
    incoming-queue incoming-queue;
    transaction-queue transaction-queue;
}

shared sic group identifier radius accounting-listener transport
shared sic group identifier radius accounting-listener transport id {
    port port;
    connections-per-thread connections-per-thread;
    connect-timeout connect-timeout;
    disconnect-timeout disconnect-timeout;
}
```

To configure the RADIUS accounting listener for the SIC group:

1. From configuration mode, access the statement that configures the RADIUS accounting listener queue limits for the SIC group.

```
[edit]
user@host# edit shared sic group group1 radius accounting-listener limit
```

2. (Optional) Specify the incoming queue limit for the RADIUS accounting listener.

```
[edit shared sic group group1 radius accounting-listener limit]
user@host# set incoming-queue incoming-queue
```

3. (Optional) Specify the transaction queue limit for the RADIUS accounting listener.

```
[edit shared sic group group1 radius accounting-listener limit]
user@host# set transaction-queue transaction-queue
```

4. Specify the name of the RADIUS accounting listener transport used to listen for accounting requests from RADIUS clients over UDP. For example, to configure the RADIUS accounting listener transport called `acctrp1` for the SIC group `group1`:

```
[edit shared sic group group1 radius accounting-listener transport]
user@host# set id acctrp1
```

5. Specify the UDP port number of the accounting listener from which the server listens for RADIUS packets.

```
[edit shared sic group group1 radius accounting-listener transport acctrp1]
user@host# set port port
```

6. (Optional) Specify the number of UDP connections per thread.

```
[edit shared sic group group1 radius accounting-listener transport acctrp1]
user@host# set connections-per-thread connections-per-thread
```

7. (Optional) Specify the UDP connection timeout in milliseconds.

```
[edit shared sic group group1 radius accounting-listener transport acctrp1]
user@host# set connect-timeout connect-timeout
```

8. (Optional) Specify the UDP disconnection timeout in milliseconds.

```
[edit shared sic group group1 radius accounting-listener transport acctrp1]
user@host# set disconnect-timeout disconnect-timeout
```

Related Topics

- Example: Basic SIC Group Configuration (SRC CLI) on page 449
- Configuring Upstream RADIUS Network Elements and Clients for the SIC Group (SRC CLI) on page 430
- Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI) on page 423
- Configuring the RADIUS Transport for a SIC Server (SRC CLI) on page 425

Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI)

You can use the RADIUS outbound transport to control communication to accounting targets that reside in a downstream network element. You need to specify the UDP port, as well as connect and disconnect related configuration options of the SIC group.

Use the following statements to configure the outbound RADIUS transport of the SIC group:

```
shared sic group identifier radius outbound-transport transport-name
```

```
shared sic group identifier radius outbound-transport transport-name {  
  connections-per-thread connections-per-thread;  
  connect-timeout connect-timeout;  
  disconnect-timeout disconnect-timeout;  
  port port;  
  port-range-size port-range-size;  
}
```

To configure the outbound RADIUS transport of the SIC group:

1. From configuration mode, access the statement that configures the name for the outbound RADIUS transport of the SIC group. For example, to configure the outbound RADIUS transport called outtrp1 for the SIC group group1:

```
[edit]  
user@host# edit shared sic group group1 radius outbound-transport outtrp1
```

2. (Optional) Specify the number of UDP connections per thread.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set connections-per-thread connections-per-thread
```

3. (Optional) Specify the UDP connection timeout in milliseconds.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set connect-timeout connect-timeout
```

4. (Optional) Specify the UDP disconnection timeout in milliseconds.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set disconnect-timeout disconnect-timeout
```

5. Specify the UDP port number starting from which the server sends the RADIUS packets.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set port port
```

6. (Optional) Specify the range of UDP ports that are used to send the RADIUS packets.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set port-range-size port-range-size
```

Related Topics

- Example: Basic SIC Group Configuration (SRC CLI) on page 449
- Overview of RADIUS Transports for the SIC Group and Server on page 407
- Configuring the RADIUS Transport for a SIC Server (SRC CLI) on page 425
- Configuring Downstream RADIUS Network Elements and Accounting Targets for the SIC Group (SRC CLI) on page 431

Configuring the RADIUS Transport for a SIC Server (SRC CLI)

You need to configure both the inbound and outbound RADIUS transport for each server in the SIC group. Servers use the same inbound and outbound transports that are configured for the SIC group.

Use the following statements to configure the RADIUS transport options for the SIC server:

```
shared sic group identifier server identifier transport transport-name
```

```
shared sic group identifier server identifier transport transport-name {  
  address address;  
}
```

```
shared sic group identifier server identifier outbound-transport transport-name
```

```
shared sic group identifier server identifier outbound-transport transport-name {  
  address address;  
}
```

To configure the RADIUS transport options for the SIC server:

1. From configuration mode, access the statement that configures the RADIUS inbound transport options for the server. For example, if the accounting listener transport for the group is configured as `trpin1`, specify the server transport as `trpin1`:

```
[edit]  
user@host# edit shared sic group group1 server server1 transport trpin1
```

2. (Optional) Specify the IP address used by the server for receiving UDP packets.

```
[edit shared sic group group1 server server1 transport trpin1]  
user@host# set address address
```

3. Specify the RADIUS outbound transport options for the SIC server. For example, if the outbound transport for the SIC group is set to `trpout1`, set the server outbound transport to `trpout1`:

```
[edit]  
user@host# edit shared sic group group1 server server1 outbound-transport trpout1
```

4. (Optional) Specify the IP address used by the server when sending outbound requests.

```
[edit shared sic group group1 server server1 outbound-transport trpout1]  
user@host# set address address
```

Related Topics

- Example: Basic SIC Group Configuration (SRC CLI) on page 449
- Overview of RADIUS Transports for the SIC Group and Server on page 407
- Configuring the RADIUS Accounting Listener for the SIC Group (SRC CLI) on page 422

- Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI) on page 423
- Adding a Server to a SIC Group (SRC CLI) on page 419

Configuring Dictionaries for the SIC Group (SRC CLI)

You can add new attributes or modify the current attributes in dictionaries that are already loaded into the shared SIC group configuration. You cannot delete attributes in a dictionary, or delete the dictionary itself. To add or modify an attribute in a dictionary, specify the unique name of the attribute, and configure the RADIUS properties of the attribute.



NOTE: To create a new dictionary, we recommend that you work with Juniper Networks Technical Support.

Use the following statements to configure attributes in SIC dictionary:

```
shared sic group identifier dictionary id
```

```
shared sic group identifier dictionary id attribute id
```

```
shared sic group identifier dictionary id attribute id radius {  
    type type;  
    format (one-byte-integer | integer | eight-byte-integer | string | ipv4-address |  
        ipv6-address | time | octets);  
    vendor-id vendor-id;  
    encrypt;  
    salt-encrypt;  
    tagged;  
    sensitive;  
}
```

```
shared sic group identifier dictionary id attribute id radius constant constant-name {  
    constant-value;  
}
```

To add or modify attributes in a SIC dictionary:

1. From configuration mode, access the statement that specifies the unique name for the dictionary. This sample procedure uses `group1` as the group identifier and `dic1` as the dictionary identifier.

```
[edit]  
user@host# edit shared sic group group1 dictionary dic1
```

2. Specify the unique name for the attribute you want to add or modify in the dictionary.

```
[edit shared sic group group1 dictionary dictionary1]  
user@host# edit attribute id
```

3. Specify that the attribute is a RADIUS attribute.

```
[edit shared sic group group1 dictionary dictionary1 attribute id]
user@host# edit radius
```

4. Specify the attribute type.

```
[edit shared sic group group1 dictionary dictionary1 attribute attribute1 radius]
user@host# set type type
```

5. Specify the format of the RADIUS attribute.

```
[edit shared sic group group1 dictionary dictionary1 attribute attribute1 radius]
user@host# set format (one-byte-integer | integer | eight-byte-integer | string |
  ipv4-address | ipv6-address | time | octets)
```

where:

- one-byte-integer—Attribute value is an 8-bit unsigned integer.
- integer—Attribute value is a 32-bit unsigned integer.
- eight-byte-integer—Attribute value is a 64-bit unsigned integer.
- string—Attribute value is a string.
- ipv4-address—Attribute value is an IPv4 address.
- ipv6-address—Attribute value is an IPv6 address.
- time—Attribute value is a 32-bit unsigned value, with the most significant octet appearing first. The value is equal to the number of seconds since 00:00:00 UTC, January 1, 1970.
- octets—Attribute value consists of raw bytes.

6. (Optional) Specify the vendor id for the attribute.

```
[edit shared sic group group1 dictionary dictionary1 attribute attribute1 radius]
user@host# set vendor-id vendor-id
```

7. (Optional) Specify whether the attribute should be encrypted without the salt.

```
[edit shared sic group group1 dictionary dictionary1 attribute attribute1 radius]
user@host# set encrypt
```

8. (Optional) Specify whether the attribute should be encrypted with the salt.

```
[edit shared sic group group1 dictionary dictionary1 attribute attribute1 radius]
user@host# set salt-encrypt
```

9. (Optional) Specify whether the RADIUS attribute is tagged.

```
[edit shared sic group group1 dictionary dictionary1 attribute attribute1 radius]
user@host# set tagged
```

10. (Optional) Specify whether the RADIUS attribute carries sensitive data, so its value will not be logged.

```
[edit shared sic group group1 dictionary dictionary1 attribute attribute1 radius]
user@host# set sensitive
```

11. (Optional) Specify the name and value of the constant you want to associate with the data contained in the RADIUS attribute.

```
[edit shared sic group group1 dictionary dictionary1 attribute attribute1 radius constant]
user@host# set constant-name constant-name constant-value
```

12. (Optional) If you modify an existing dictionary, you need to restart the SIC.

```
user@host# restart component sic
```

13. (Optional) Load the dictionary into the group configuration. This step is required only if the dictionary is not currently loaded in the group. See “Loading a Dictionary into the SIC Shared Group Configuration (SRC CLI)” on page 428.

Related Topics

- Example: Basic SIC Group Configuration (SRC CLI) on page 449
- Configuring the Device Models Supported by the SIC Group (SRC CLI) on page 429
- Overview of SIC Dictionaries and Device Models (SRC CLI) on page 407
- Overview of Loading SIC Dictionaries (SRC CLI) on page 408

Loading a Dictionary into the SIC Shared Group Configuration (SRC CLI)

Whenever you create a new SIC group, you must assign it a dictionary by loading the dictionary into the shared SIC group configuration.

To load a dictionary into a new SIC group:

1. Navigate to the dictionary. For example, to load the default dictionary called radius into a new SIC group called g2:

```
[edit shared sic group g2 dictionary]
user@host# load radius dictionary radius
Load Dictionary Complete
```

2. (Optional) If you are loading a dictionary that has the same name as a currently loaded dictionary, you need to define how you want to merge the two dictionaries by setting the merge option. You can either merge the two dictionaries, or you can replace the existing dictionary with the new dictionary of the same name. For example, to merge a dictionary called dic1 into the SIC group g2 which already contains a dictionary called dic1:

```
[edit shared sic group g2 dictionary]
user@host# load radius dictionary dic1 merge-option merge
Load Dictionary Complete
```



CAUTION: Use caution when loading dictionaries into the shared SIC group configuration. You can merge the dictionaries, or replace the existing dictionary. If you set the merge option to replace, be aware that all contents in the existing dictionary are replaced by the contents of the new dictionary.

It is necessary to load a dictionary only if you create a new shared SIC group configuration. The SIC will not work without a dictionary. If the SIC group already has a dictionary loaded and you want to add or modify an attribute, edit the dictionary by using the `shared sic group identifier dictionary` configuration statement.

Related Topics

- Configuring Dictionaries for the SIC Group (SRC CLI) on page 426
- Example: Basic SIC Group Configuration (SRC CLI) on page 449
- Configuring the Device Models Supported by the SIC Group (SRC CLI) on page 429
- Overview of SIC Dictionaries and Device Models (SRC CLI) on page 407
- Overview of Loading SIC Dictionaries (SRC CLI) on page 408

Configuring the Device Models Supported by the SIC Group (SRC CLI)

To configure the device models supported by the SIC group:

1. From configuration mode, access the statement that configures the device models supported by the SIC group. For example, to configure the device associated with the model name `dm1` for the group `group1`:

```
[edit]
user@host# edit shared sic group group1 model dm1
```

2. Specify the name of the dictionary used by the device model.

```
[edit shared sic group group1 model dm1]
user@host# set dictionary dictionary
```

Related Topics

- Overview of SIC Dictionaries and Device Models (SRC CLI) on page 407
- Configuring Dictionaries for the SIC Group (SRC CLI) on page 426
- Configuring Downstream RADIUS Network Elements and Accounting Targets for the SIC Group (SRC CLI) on page 431
- Configuring Upstream RADIUS Network Elements and Clients for the SIC Group (SRC CLI) on page 430
- Example: Basic SIC Group Configuration (SRC CLI) on page 449

Configuring Upstream RADIUS Network Elements and Clients for the SIC Group (SRC CLI)

Upstream network elements contain one or more SIC accounting clients, which send accounting requests to the SIC server. To receive packets from an accounting client, you need to configure the upstream network element and the accounting client properties.

Use the following statements to configure the upstream RADIUS network elements and clients for the SIC group:

```
shared sic group identifier radius network-element id

shared sic group identifier radius network-element id upstream {
    model model;
}
shared sic group identifier radius network-element id upstream accounting-client id {
    address address;
    accounting-secret accounting-secret;
}
```

To configure upstream RADIUS network elements and clients for the SIC group:

1. From configuration mode, access the statement that configures the RADIUS network element. For example, to configure the upstream RADIUS network element called *ne1* for the SIC group *group1*:

```
[edit]
user@host# edit shared sic group group1 radius network-element ne1
```

2. Specify the network element as being upstream from the SIC server, and specify a device model for the accounting. The device model must have previously been configured for the SIC group.

```
[edit shared sic group group1 radius network-element ne1 upstream]
user@host# set model model
```

3. Specify the name of the RADIUS accounting client.

```
[edit shared sic group group1 radius network-element ne1 upstream accounting-client]
user@host# set id id
```

4. Specify the IP address of RADIUS accounting client sending accounting requests to the SIC server.

```
[edit shared sic group group1 radius network-element ne1 upstream accounting-client]
user@host# set address address
```

5. Specify the shared secret used by the accounting client.

```
[edit shared sic group group1 radius network-element ne1 upstream accounting-client]]
user@host# set accounting-secret accounting-secret
```

- Related Topics**
- Overview of RADIUS Configuration for the SIC (SRC CLI) on page 404
 - Overview of RADIUS Transports for the SIC Group and Server on page 407
 - Configuring the Device Models Supported by the SIC Group (SRC CLI) on page 429
 - Configuring the RADIUS Accounting Listener for the SIC Group (SRC CLI) on page 422
 - Configuring the RADIUS Transport for a SIC Server (SRC CLI) on page 425
 - Overview of SIC Dictionaries and Device Models (SRC CLI) on page 407
 - Example: Basic SIC Group Configuration (SRC CLI) on page 449

Configuring Downstream RADIUS Network Elements and Accounting Targets for the SIC Group (SRC CLI)

You can configure downstream RADIUS network elements, which contain one or more AAA server accounting targets to which the SIC forwards RADIUS accounting requests using the RADIUS protocol.

Configuring downstream network elements and accounting targets involves the following tasks:

1. Configuration Statements for Downstream Network Elements and Accounting Targets on page 431
2. Configuring the Downstream Network Element and Device Model on page 432
3. Configuring SIC Accounting Targets (SRC CLI) on page 433
4. Configuring the Failover Policy for the Network Element (SRC CLI) on page 433
5. Configuring the Fast Fail Options for the Failover Policy on page 434
6. Configuring the Retry Options for the Failover Policy on page 434

Configuration Statements for Downstream Network Elements and Accounting Targets

Use the following statements to configure downstream RADIUS network elements and accounting targets for the SIC group:

```
shared sic group identifier radius network-element id
```

```
shared sic group identifier radius network-element id downstream {  
    model model;  
}
```

```
shared sic group identifier radius network-element id downstream {  
    failover-mode (round-robin | primary-backup);  
}
```

```
shared sic group identifier radius network-element id downstream accounting-target {  
    name name;  
    address address;  
}
```

```
shared sic group identifier radius network-element id downstream accounting-target name
{
    secret secret;
    outbound-transport outbound-transport;
    port port;
}

shared sic group identifier radius network-element id downstream failover-policy {
    priority priority;
}

shared sic group identifier radius network-element id downstream failover-policy fast-fail
{
    minimum-number minimum-number;
    timeout timeout;
    reset-delay reset-delay;
}

shared sic group identifier radius network-element id downstream failover-policy retry {
    number number;
    timeout timeout;
}
```

Configuring the Downstream Network Element and Device Model

To configure the downstream RADIUS network element and the device model:

1. From configuration mode, access the statement that configures the RADIUS network element, and specify it as a downstream network element. This sample procedure uses group1 for the SIC group, ne1 for the network element identifier.

[edit]

```
user@host# edit shared sic group group1 radius network-element ne1 downstream
```

2. Specify the failover mode used by the SIC server for the downstream network element.

[edit shared sic group group1 radius network-element ne1 downstream]

```
user@host# set failover-mode (round-robin | primary-backup)
```

where:

- **round-robin**—Failover mode used by the server is the round robin method. When this failover mode is used, the server alternates the path it uses to send messages to the downstream RADIUS network element target.
 - **primary-backup**—Failover mode used by the server is the primary backup method. When this failover mode is used, the server sends requests to the primary target unless it is unavailable, in which case it sends requests to the backup target.
3. (Optional) and specify a device model for the accounting target. The device model must have previously been configured for the SIC group.

[edit shared sic group group1 radius network-element ne1 downstream]

```
user@host# set model model
```


Configuring SIC Accounting Targets (SRC CLI)

To configure the accounting target:

1. From configuration mode, access the statement that configures the accounting target. This sample procedure uses `group1` for the SIC group, `ne1` for the network element identifier, and `target1` as the accounting target.

```
edit shared sic group group1 radius network-element ne1 downstream  
accounting-target target1
```

2. Specify the IP address of the RADIUS accounting target (AAA server) contained in the network element.

```
[edit shared sic group group1 radius network-element ne1 downstream  
accounting-target target1]  
user@host# set address address
```

3. Specify the shared secret used by the RADIUS accounting target.

```
[edit shared sic group group1 radius network-element ne1 downstream  
accounting-target target1]  
user@host# set secret secret
```

4. (Optional) Specify the name of the local transport on the SIC server sending outbound requests to the downstream accounting target.

```
[edit shared sic group group1 radius network-element ne1 downstream  
accounting-target target1]  
user@host# set outbound-transport outbound-transport
```

5. (Optional) Specify the UDP port number on which the RADIUS accounting target listens for accounting requests.

```
[edit shared sic group group1 radius network-element ne1 downstream  
accounting-target target1]  
user@host# set port port
```

Configuring the Failover Policy for the Network Element (SRC CLI)

To configure the failover policy:

1. From configuration mode, access the statement that configures the failover policy. This sample procedure uses `group1` as the SIC group and `ne1` as the network element identifier.

```
user@host# edit shared sic group group1 radius network-element ne1 downstream  
failover-policy
```

2. Specify the failover policy priority. Targets with lower priority values are selected before other targets in a failover policy.

```
[edit shared sic group group1 radius network-element ne1 downstream failover-policy]  
user@host# set priority priority
```

Configuring the Fast Fail Options for the Failover Policy

To configure the fast fail options for the failover policy:

1. From configuration mode, access the statement that configures the fast fail options for failover policy. This sample procedure uses group1 as the SIC group and ne1 as the network element identifier.

```
edit shared sic group group1 radius network-element ne1 downstream failover-policy fast-fail
```

2. Specify the minimum number of times the server will retransmit a message if an acknowledgment from the network element is not received.

```
[edit shared sic group group1 radius network-element ne1 downstream failover-policy fast-fail]  
user@host# set minimum-number minimum-number
```

3. Specify the time in seconds before the SIC server goes into fast fail mode for that accounting target.

```
[edit shared sic group group1 radius network-element ne1 downstream failover-policy fast-fail]  
user@host# set timeout timeout
```

4. Specify the time in seconds after which the SIC server comes out of fast fail mode for that accounting target.

```
[edit shared sic group group1 radius network-element ne1 downstream failover-policy fast-fail]  
user@host# set reset-delay reset-delay
```

Configuring the Retry Options for the Failover Policy

To configure the retry options for the failover policy:

1. From configuration mode, access the statement that configures the retry options for failover policy. This sample procedure uses group1 as the SIC group and ne1 as the network element identifier.

```
edit shared sic group group1 radius network-element ne1 downstream failover-policy retry
```

2. Specify the maximum number of times a message is retransmitted if an acknowledgment from the network element is not received.

```
[edit shared sic group group1 radius network-element ne1 downstream failover-policy retry]  
user@host# set number number
```

3. Specify the number of seconds between retry attempts.

```
[edit shared sic group group1 radius network-element ne1 downstream failover-policy retry]
```

```
user@host# set timeout timeout
```

- Related Topics**
- Overview of RADIUS Configuration for the SIC (SRC CLI) on page 404
 - Overview of RADIUS Transports for the SIC Group and Server on page 407
 - Configuring the Device Models Supported by the SIC Group (SRC CLI) on page 429
 - Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI) on page 423
 - Configuring Implicit Accounting Routes for the SIC (SRC CLI) on page 444
 - Configuring Explicit Accounting Routes for the SIC (SRC CLI) on page 442
 - Example: Basic SIC Group Configuration (SRC CLI) on page 449

Configuring What Realms Are Local to the SIC Group (SRC CLI)

To configure what realms are local to the SIC group:

1. From configuration mode, access the statement that configures local realms. For example, to configure the local realm called *realm1* for the group *group1*:

```
[edit]  
user@host# edit shared sic group group1 local-realm realm1
```

- Related Topics**
- Overview of SIC Local Realms on page 409
 - Overview of Local and Shared Configurations for the SIC (SRC CLI) on page 394
 - Creating a SIC Group (SRC CLI) on page 417
 - Creating a SIC Server Instance (SRC CLI) on page 419

Configuration Statements for SIC Editing Rules (SRC CLI)

Use the following statements to configure the optional SIC editing rules at the **[edit]** hierarchy level.

Use the following statements to create the editing rule and specify the type of source used in the editing rule:

```
shared sic group identifier editing editing-rule {  
    mode (replace | append);  
}  
shared sic group identifier editing editing-rule default {  
    literal literal;  
    request-attribute request-attribute;  
    variable variable;  
}
```

Use the following statements to configure the editing rule when you specify a literal as the source of the editing rule:

```
shared sic group identifier editing editing-rule source literal
shared sic group identifier editing editing-rule source literal identifier condition realm {
    (present | not-present);
}
shared sic group identifier editing editing-rule source literal identifier condition realm
    does-not-equal value
shared sic group identifier editing editing-rule source literal identifier condition realm equals
    value
shared sic group identifier editing editing-rule source literal identifier condition realm
    has-prefix value
shared sic group identifier editing editing-rule source literal identifier condition realm
    has-suffix value
shared sic group identifier editing editing-rule source literal identifier condition realm range
{
    low low;
    high high;
}
shared sic group identifier editing editing-rule source literal identifier condition request {
}
shared sic group identifier editing editing-rule source literal identifier condition request
    attribute attribute-name {
        (present | not-present);
    }
shared sic group identifier editing editing-rule source literal identifier condition request
    attribute attribute-name does-not-equal value
shared sic group identifier editing editing-rule source literal identifier condition request
    attribute attribute-name equals value
shared sic group identifier editing editing-rule source literal identifier condition request
    attribute attribute-name has-prefix value
shared sic group identifier editing editing-rule source literal identifier condition request
    attribute attribute-name has-suffix value
shared sic group identifier editing editing-rule source literal identifier condition request
    attribute attribute-name range {
        low low;
        high high;
    }
}
shared sic group identifier editing editing-rule source literal identifier condition user-identity
{
    (present | not-present);
}
shared sic group identifier editing editing-rule source literal identifier condition user-identity
    does-not-equal value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
    equals value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
    has-prefix value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
    has-suffix value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
    range {
        low low;
        high high;
    }
}
```

Use the following statements to configure the editing rule when you specify a request attribute as the source of the editing rule:

```

shared sic group identifier editing editing-rule source request-attribute identifier {
    remove-prefix remove-prefix;
    remove-suffix remove-suffix;
    remove-before remove-before;
    remove-after remove-after;
}
shared sic group identifier editing editing-rule source request-attribute identifier condition
    realm {
        (present | not-present);
    }
shared sic group identifier editing editing-rule source request-attribute identifier condition
    realm does-not-equal value
shared sic group identifier editing editing-rule source request-attribute identifier condition
    realm equals value
shared sic group identifier editing editing-rule source request-attribute identifier condition
    realm has-prefix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
    realm has-suffix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
    realm range {
        low low;
        high high;
    }
shared sic group identifier editing editing-rule source request-attribute identifier condition
    request {
    }
shared sic group identifier editing editing-rule source request-attribute identifier condition
    request attribute attribute-name {
        (present | not-present);
    }
shared sic group identifier editing editing-rule source request-attribute identifier condition
    request attribute attribute-name does-not-equal value
shared sic group identifier editing editing-rule source request-attribute identifier condition
    request attribute attribute-name equals value
shared sic group identifier editing editing-rule source request-attribute identifier condition
    request attribute attribute-name has-prefix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
    request attribute attribute-name has-suffix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
    request attribute attribute-name range {
        low low;
        high high;
    }
shared sic group identifier editing editing-rule source request-attribute identifier condition
    user-identity {
        (present | not-present);
    }
shared sic group identifier editing editing-rule source request-attribute identifier condition
    user-identity does-not-equal value
shared sic group identifier editing editing-rule source request-attribute identifier condition
    user-identity equals value
shared sic group identifier editing editing-rule source request-attribute identifier condition
    user-identity has-prefix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
    user-identity has-suffix value

```

```
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity range {
    low low;
    high high;
  }
```

Use the following statements to configure the editing rule when you specify a SIC variable as the source of the editing rule:

```
shared sic group identifier editing editing-rule source variable identifier
shared sic group identifier editing editing-rule source variable identifier condition realm {
  (present | not-present);
}
shared sic group identifier editing editing-rule source variable identifier condition realm
  does-not-equal value
shared sic group identifier editing editing-rule source variable identifier condition realm
  equals value
shared sic group identifier editing editing-rule source variable identifier condition realm
  has-prefix value
shared sic group identifier editing editing-rule source variable identifier condition realm
  has-suffix value
shared sic group identifier editing editing-rule source variable identifier condition realm
  range {
    low low;
    high high;
  }
shared sic group identifier editing editing-rule source variable identifier condition request {
}
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name {
    (present | not-present);
  }
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name does-not-equal value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name equals value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name has-prefix value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name has-suffix value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name range {
    low low;
    high high;
  }
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  {
    (present | not-present);
  }
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  does-not-equal value
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  equals value
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  has-prefix value
```

```

shared sic group identifier editing editing-rule source variable identifier condition user-identity
  has-suffix value
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  range {
    low low;
    high high;
  }

```

Use the following statements to configure target of the editing rule:

```

shared sic group identifier editing editing-rule target {
  request-attribute request-attribute;
  response-attribute response-attribute;
  variable variable;
}

```

- Related Topics**
- SIC Editing Rules Overview (SRC CLI) on page 401
 - Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI) on page 439
 - Configuring Explicit Accounting Routes for the SIC (SRC CLI) on page 442
 - Example: Basic SIC Group Configuration (SRC CLI) on page 449

Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI)

When you use explicit routing for the SIC, you can optionally specify an editing rule you want applied to the accounting request before SIC sends the request to the accounting target. To configure editing rules, you define a source, conditions, and a target.

Table 30 on page 439 lists the available sources, conditions, and targets you can define in editing rules, and “Configuration Statements for SIC Editing Rules (SRC CLI)” on page 435 provides a complete list of configuration statements used to define editing rules.

Table 30: SIC Editing Rule Options

Source	Conditions	Target
SIC literal	Match conditions:	Transactional variable
Transactional variable	<ul style="list-style-type: none"> • Realm • User identity 	RADIUS attribute in the request
RADIUS attribute in the request	<ul style="list-style-type: none"> • Request attribute 	RADIUS attribute in the response
	Condition tests:	
	<ul style="list-style-type: none"> • Present • Not present • Equals • Does not equal • Has suffix • Has prefix • Within range 	

To configure an editing rule:

1. From configuration mode, access the statement that configures the editing rule, and specify a name for the editing rule. For example, to create an editing rule called `er1`:

```
[edit]
user@host# edit shared sic group identifier editing er1
```

2. Specify the editing mode.

```
[edit shared sic group identifier editing er1]
user@host# set mode (replace | append)
```

Where:

- **replace**—Current target (LValue) is replaced with the new value from the editing process
- **append**—Current target (LValue) value is concatenated with the new target value from the editing process

3. Define the source of the editing rule. The source can be either a literal, transactional variable, or an attribute in the request. For example, to define a literal called `literal1` as the source:

```
[edit]
edit shared sic group identifier editing er1 source literal
user@host# set literal1
```

4. (Optional) If the source is a request attribute, you can also specify whether to remove the prefix, remove the suffix, remove before @, or remove after @.

```
[edit ]
user@host# edit shared sic group identifier editing er1 source request-attribute
identifier
user@host# set remove-prefix remove-prefix
```

5. Define the editing rule conditions, which include specifying the match conditions and the condition tests. See Table 30 on page 439 and “Configuration Statements for SIC Editing Rules (SRC CLI)” on page 435 for a complete list of configuration statements used to specify SIC editing rules. For example, to specify a condition that examines literals in accounting requests for the realm=`abc.com`:

```
[edit]
edit shared sic group identifier editing er1 literal literal1 condition realm equals
user@host# set abc.com
```

6. Define the target (where you want the result of the editing process to be placed) of the editing rule. The target can either be a transactional variable, a RADIUS attribute in the request, or a RADIUS attribute in the response. For example, to place the results of the editing process in a variable called `sic-variable1`:

```
[edit]
user@host# edit shared sic group identifier editing er1 target
user@host# set variable sic-variable1
```


7. (Optional) Specify a default editing rule. You can set default editing rules for all three source types (literal, variable, request attribute).

[edit]

user@host# **edit** shared sic group identifier editing editing-rule default

user@host# **set** literal *literal*

Related Topics

- SIC Editing Rules Overview (SRC CLI) on page 401
- Configuring Explicit Accounting Routes for the SIC (SRC CLI) on page 442
- Overview of SIC Accounting Methods and Targets (SRC CLI) on page 396
- Configuration Statements for SIC Editing Rules (SRC CLI) on page 435
- Configuration Statements for SIC Explicit Accounting Routing Rules on page 441
- Overview of SIC Request Routing Rules (SRC CLI) on page 399
- Example: Basic SIC Group Configuration (SRC CLI) on page 449

Configuration Statements for SIC Explicit Accounting Routing Rules

Use the following statements to configure explicit routing rules for the SIC at the [edit] hierarchy level:

```
shared sic group identifier server identifier accounting-route
shared sic group identifier server identifier accounting-route id condition realm {
  (present | not-present);
}
shared sic group identifier server identifier accounting-route id condition realm
  does-not-equal value
shared sic group identifier server identifier accounting-route id condition realm equals value
shared sic group identifier server identifier accounting-route id condition realm has-prefix
  value
shared sic group identifier server identifier accounting-route id condition realm has-suffix
  value
shared sic group identifier server identifier accounting-route id condition realm range {
  low low;
  high high;
}
shared sic group identifier server identifier accounting-route id condition request
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name {
    (present | not-present);
  }
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name does-not-equal value
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name equals value
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name has-prefix value
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name has-suffix value
```

```
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name range {
    low low;
    high high;
  }
shared sic group identifier server identifier accounting-route id condition user-identity {
  (present | not-present);
}
shared sic group identifier server identifier accounting-route id condition user-identity
  does-not-equal value
shared sic group identifier server identifier accounting-route id condition user-identity equals
  value
shared sic group identifier server identifier accounting-route id condition user-identity
  has-prefix value
shared sic group identifier server identifier accounting-route id condition user-identity
  has-suffix value
shared sic group identifier server identifier accounting-route id condition user-identity range
  {
    low low;
    high high;
  }
shared sic group identifier server identifier accounting-route id editing editing-rule
shared sic group identifier server identifier accounting-route id target {
  accounting-method accounting-method;
}
```

- Related Topics**
- Overview of SIC Request Routing Rules (SRC CLI) on page 399
 - SIC Editing Rules Overview (SRC CLI) on page 401
 - Configuring Explicit Accounting Routes for the SIC (SRC CLI) on page 442
 - Overview of SIC Accounting Methods and Targets (SRC CLI) on page 396
 - Example: Basic SIC Group Configuration (SRC CLI) on page 449

Configuring Explicit Accounting Routes for the SIC (SRC CLI)

Explicit accounting routing rules can be specified based on any combination of the conditions listed in Table 31 on page 443. For a complete list of statements used to configure explicit accounting routing rules, see “Configuration Statements for SIC Explicit Accounting Routing Rules” on page 441.

Table 31: Explicit Routing Rule Conditions

Match Condition	Condition Tests
<ul style="list-style-type: none"> • Realm • User identity • Request attribute 	<ul style="list-style-type: none"> • Present • Not present • Equals • Does not equal • Has suffix • Has prefix • Within range

To configure explicit accounting routes:

1. From configuration mode, access the configuration statement used to configure explicit accounting routes. For example, to configure a route called route66 for the server svr1, in a group called g1:

```
[edit]
user@host# edit shared sic group g1 server svr1 accounting-route route66
```

2. Define the conditions associated with the explicit accounting route. See Table 31 on page 443 to identify the conditions you want to use, and see “Configuration Statements for SIC Explicit Accounting Routing Rules” on page 441 for a list of the respective configuration statements you need to use to define the conditions. For example, to configure a condition that examines the request for a realm=abc.com:

```
[edit]
user@host# edit shared sic group g1 server svr1 accounting-route route66 condition
realm
user@host# set abc.com
```

3. (Optional) Specify the editing rule you want applied to the request before sending the request to the accounting target.

```
[edit]
user@host# edit shared sic group g1 server svr1 accounting-route route66 editing
editing-rule
```

4. Configure the accounting target by specifying the name of a defined accounting method.

```
[edit]
user@host# shared sic group g1 server svr1 accounting-route route66 target
user@host# set accounting-method accounting-method
```

Related Topics

- Overview of SIC Request Routing Rules (SRC CLI) on page 399
- Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI) on page 439
- SIC Editing Rules Overview (SRC CLI) on page 401
- Overview of SIC Accounting Methods and Targets (SRC CLI) on page 396

- Example: Basic SIC Group Configuration (SRC CLI) on page 449

Configuring Implicit Accounting Routes for the SIC (SRC CLI)

You configure implicit accounting routes by specifying the name of a previously configured network element that has the proxy function assigned to it. You can also define either a default route used for all requests from all realms, or you can specify that only requests from specific realms are routed to the proxy AAA server. When you specify specific realms, you have the option to set a condition of either an exact match of the realm string, or an match on the prefix of the realm string.

Use the following statements to configure implicit routes for the SIC:

```
shared sic group identifier radius network-element id proxy {  
}  
shared sic group identifier radius network-element id proxy realm realmValue {  
condition (exact | prefix);  
}
```

To configure implicit accounting routes for the SIC:

1. From configuration mode, access the statement that configures the remote AAA server as a proxy. For example, to configure the AAA server in a network element called `ne1` as a proxy:

```
[edit]  
user@host# edit shared sic group group1 radius network-element ne1 proxy
```

2. (Optional) Specify that only requests from specific realms are routed to the proxy AAA server, by specifying the names of the realms. For example, to specify that all requests from the realm called `abc.com` are routed to the proxy AAA server:

```
[edit shared sic group group1 radius network-element ne1 proxy]  
user@host# edit realms abc.com
```

3. (Optional) Specify the match condition for the realm.

```
[edit shared sic group group1 radius network-element ne1 proxy realm abc.com]  
user@host# set condition exact | prefix
```

4. (Optional) Specify whether you want this proxy AAA server to be the default route for requests from all realms.

```
[edit shared sic group group1 radius network-element ne1 proxy]  
user@host# set default-route-for-all-realms
```

Related Topics

- Overview of SIC Request Routing Rules (SRC CLI) on page 399
- Overview of RADIUS Configuration for the SIC (SRC CLI) on page 404
- Configuring Downstream RADIUS Network Elements and Accounting Targets for the SIC Group (SRC CLI) on page 431

- Overview of SIC Accounting Methods and Targets (SRC CLI) on page 396
- Example: Basic SIC Group Configuration (SRC CLI) on page 449

Configuring Event Logging for a SIC Server (SRC CLI)

You can configure the SIC server to capture any number of log streams called loggers. If you configure multiple log streams, make sure you configure unique names for each log stream. You can configure the log stream to display only log messages from particular log groups. To configure the event level for a log group, you first specify the log group and then specify the event level for it.

Use the following statements to configure event logging for the SIC server:

```
shared sic group identifier server identifier

shared sic group identifier server identifier logger id

shared sic group identifier server identifier logger id file {
  filter (/error | /debug-error);
  filename filename;
  maximum-file-size maximum-file-size;
  rollover-interval rollover-interval;
  rollover-on-startup;
  flush-after-writes;
  high-resolution-timestamps;
  header header;
  footer footer;
  prepend-message-header;
  work-id-label work-id-label;
  work-id-padding work-id-padding;
  utc;
}
shared sic group identifier server identifier logger id group (administration | configuration
| system | packet | packet-trace | packet-trace-raw) {
  events (error | warning | standard | detail | debug);
}
```

To configure event logging for the SIC server:

1. From configuration mode, access the statement that configures the server belonging to the SIC group. For example, to configure the server called `sicscr1` for the group `group1`:

```
[edit]
user@host# edit shared sic group group1 server sicscr1
```

2. Specify the name used by the server to identify the log stream.

```
[edit shared sic group group1 server sicscr1 logger]
user@host# set id log1
```

3. (Optional) Specify the filter to define which event messages are logged or ignored.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set filter (/error | /debug-error)
```

where:

- **/error**—Error events are captured for every log group
- **/debug-error**—Debug events are captured for every log group

4. Specify the prefix to be added to the log file for easy identification.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set filename filename
```

5. (Optional) Specify the maximum size of the log file and the rollover file.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set maximum-file-size maximum-file-size
```

6. (Optional) Specify the time in seconds for the rollover interval after which the new log file is created.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set rollover-interval rollover-interval
```

7. (Optional) Specify whether the new log file is to be created every time the server starts.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set rollover-on-startup
```

8. (Optional) Specify whether or not to buffer log messages.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set flush-after-writes
```

- If set, log messages are immediately written to the log file without buffering. Use this setting for real-time logging.
- If not set, SIC log messages are kept in the buffer until the buffer is full and then all messages in the buffer are written to the log file. Use this setting for performance optimization, when real-time logging is not needed.

9. (Optional) Specify whether the high resolution time reporting system functions are used.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set high-resolution-timestamps
```

10. (Optional) Specify the header message to be added to the beginning of each log file.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set header header
```

11. (Optional) Specify the footer message to be added to the end of each log file.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set footer footer
```
12. (Optional) Specify whether to prepend each log message with additional information such as time, thread, and transaction information.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set prepend-message-header
```
13. (Optional) Specify the work data ID prefix to be added to each log message.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set work-id-label work-id-label
```
14. (Optional) Specify the string to be added to each log message if work data is not available.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set work-id-padding work-id-padding
```
15. (Optional) Specify the time and date values to Universal Time Coordinates (UTC, formerly known as Greenwich Mean Time, or GMT). If disabled, time and date reflect local time.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set utc
```
16. Configure the event level for each log group for which you want to collect events. First specify the name of the log group, and then specify the event level. Repeat the process for each log group for which you want to collect events.

```
[edit]
user@host# edit shared sic group group1 server sicser1 logger log1 group
(administration | configuration | system | packet | packet-trace | packet-trace-raw)
```

where:

- **administration**—Log group reports events related to server administration.
- **configuration**—Log group reports events related to server configuration.
- **system**—Log group reports events related to the system, such as system start and system stop.
- **packet**—Log group reports events related to transaction processing, such as incoming and outgoing packets.
- **packet-trace**—Log group displays contents of a packet. The format is attribute name:attribute value.
- **packet-trace-raw**—Log group displays raw data (octets) of incoming and outgoing packets.

17. (Optional) Specify the highest event level for the log group.

```
[edit shared sic group group1 server sicser1 logger log1 group]
user@host# set events (error | warning | standard | detail | debug)
```

where:

- **error**—Messages in log shown at event level error.
- **warning**—Messages in log shown at event levels error and warning.
- **standard**—Messages in log shown at event levels error, warning, and standard.
- **detail**—Messages in log shown at event levels error, warning, standard, and detail.
- **debug**—Messages in log shown at event levels error, warning, standard, detail, and debug.

- Related Topics**
- Overview of SIC Event Logging (SRC CLI) on page 410
 - Configuring SNMP for the SIC Group (SRC CLI) on page 448
 - Overview of Local and Shared Configurations for the SIC (SRC CLI) on page 394
 - Example: Basic SIC Group Configuration (SRC CLI) on page 449

Configuring SNMP for the SIC Group (SRC CLI)

You can configure each SNMP event and associated dilution factor. When an event occurs, an SNMP trap is sent to the SNMP manager.

Use the following statements to configure SNMP for the SIC server:

```
shared sic group identifier snmp event (sic-server-startup | sic-server-shutdown |
  sic-server-unauthorized-administration-request | sic-server-internal-error |
  sic-server-resource-failure | sic-server-log-file-failure) {
  dilution-factor dilution-factor;
}
```

To configure SNMP events for the SIC group:

1. Specify the SNMP trap name for which you want to configure the dilution factor.

```
[edit]
user@host# edit shared sic group group1 snmp event (sic-server-startup |
  sic-server-shutdown | sic-server-unauthorized-administration-request |
  sic-server-internal-error | sic-server-resource-failure | sic-server-log-file-failure)
```

where:

- **sic-server-startup**—SNMP trap on server startup.
- **sic-server-shutdown**—SNMP trap on server shutdown.

- **sic-server-unauthorized-administration-request**—SNMP trap on unauthorized administration request.
 - **sic-server-internal-error**—SNMP trap on server internal error.
 - **sic-server-resource-failure**—SNMP trap on server resource failure.
 - **sic-server-log-file-failure**—SNMP trap on server log file failure.
2. (Optional) Specify the dilution factor. The event is sent to the SNMP manager every *n* occurrences of the condition that generated the alert.

```
[edit shared sic group group1 snmp event]
user@host# set dilution-factor dilution-factor
```

- Related Topics**
- Overview of SNMP Support for the SIC (SRC CLI) on page 413
 - Configuring Event Logging for a SIC Server (SRC CLI) on page 445

Example: Basic SIC Group Configuration (SRC CLI)

This sample configuration uses the default SIC group called default-group, and the default SIC server called default-server.

An editing rule called username specifies that if the source, which is the request attribute User-Name, contains the @test.com suffix, the suffix is to be removed, and the resulting value placed in the target, which is the request attribute User-Name. A second editing rule, called vpnid, specifies that the target, which is the SIC variable vpn-id, should be replaced with the value of the source, which is the request attribute NAS-Identifier.

The SIC group (default-group) includes the default device model called default-model, which are both using the default dictionary called radius.

The accounting listener for the SIC listens on port 1813 for incoming accounting events. An upstream network element called netpc is using the default device model called default-model. The netpc network element contains four accounting clients called netpc13, netpc14, netpc15 and netpc16. The IP addresses and shared secrets of these accounting clients are provided as examples only. The outbound transport uses port 0.

The accounting route called test-route specifies that the editing rule called vpnid is to be applied before the request is routed to the accounting target, which by default is the SSR database (default-method).

Table 32 on page 449 lists the attribute mapping defined between the SIC and the SAE plug-in attributes.

Table 32: Sample Configuration Attribute Associations

SIC Variable or Attribute	SAE Plug-In Attribute
Request-attribute User-Name	Login-name

Table 32: Sample Configuration Attribute Associations (*continued*)

SIC Variable or Attribute	SAE Plug-In Attribute
Request-attribute Calling-Station-Id	Property.calling-station-id
Variable ReceiveTime	Property.session-start-time
Variable UserStatusType	Property.session-state
Request-attribute Framed-IP-Address	User-inet-address

Three log streams are configured, including the default log stream called default-logger, which captures events for the log groups at the event levels listed in Table 33 on page 450.

Table 33: Log Groups and Associated Event Level for Log Stream=default logger

Log Group	Event Level
Administration	Warning
Configuration	Warning
Packet	Debug
PacketTrace	Warning
PacketTraceRaw	Warning
System	Warning

Two additional log streams are configured, called debug-logger and error-logger, which capture events for the log groups at the event levels listed in Table 34 on page 450 and Table 35 on page 451.

Table 34: Log Groups and Associated Event Level for Log Stream=debug-logger

Log Group	Event Level
Administration	Debug
Configuration	Debug
Packet	Debug
PacketTrace	Debug
PacketTraceRaw	Debug
System	Debug

Table 35: Log Groups and Associated Event Level for Log Stream=error-logger

Log Group	Event Level
Administration	Warning
Configuration	Warning
Packet	Warning
PacketTrace	Warning
PacketTraceRaw	Warning
System	Warning

```

user@host# show slot 0 sic

initial {
  directory-connection {
    credentials *****;
    entry-dn l=SIC,ou=staticConfiguration,ou=Configuration,o=Management,o=umc;
    filter (objectClass=*);
    port 389;
    principal cn=umcadmin,o=umc;
    url 127.0.0.1;
  }
}
server {
  name default-server;
}
user@host# show shared sic group default-group accounting-method
default-method database {
  plug-in-attribute {
    login-name {
      request-attribute User-Name;
    }
    property.calling-station-id {
      request-attribute Calling-Station-Id;
    }
    property.session-start-time {
      variable ReceiveTime;
    }
    property.session-state {
      variable UserStatusType;
    }
    user-inet-address {
      request-attribute Framed-IP-Address;
    }
    vpn-id;
  }
}

[edit]

*****

```

```
user@host# show shared sic group default-group editing
username {
  mode replace;
  source {
    request-attribute {
      User-Name {
        remove-suffix @test.com;
      }
    }
  }
  target {
    request-attribute User-Name;
  }
}
vpnid {
  mode replace;
  source {
    request-attribute {
      NAS-Identifier;
    }
  }
  target {
    variable vpn-id;
  }
}

[edit]
```

```
user@host# show shared sic group default-group radius
accounting-listener {
  transport {
    1813 {
      connect-timeout 1000;
      connections-per-thread 15;
      disconnect-timeout 1000;
      port 1813;
    }
  }
}
network-element netpc {
  upstream {
    model default-model;
    accounting-client {
      netpc13 {
        accounting-secret secret;
        address 10.227.6.213;
      }
      netpc14 {
        accounting-secret secret;
        address 10.227.6.214;
      }
      netpc15 {
        accounting-secret secret;
        address 10.227.6.215;
      }
      netpc16 {
        accounting-secret secret;
      }
    }
  }
}
```

```

        address 10.227.6.216;
    }
}
}
outbound-transport {
    default-outbound-transport {
        connect-timeout 1000;
        connections-per-thread 15;
        disconnect-timeout 1000;
        port 0;
    }
}

[edit]
user@host# show shared sic group default-group dictionary radius
attribute ARAP-Challenge-Response {
    radius {
        format octets;
        type 84;
    }
}
attribute ARAP-Features {
    radius {
        format octets;
        type 71;
    }
}
attribute ARAP-Password {
    radius {
        format octets;
        type 70;
    }
}
attribute Proxy-State {
    radius {
        format string;
        type 33;
    }
}
attribute Reply-Message {
    radius {
        format string;
        type 18;
    }
}
attribute Service-Type {
    radius {
        constant Administrative {
            6;
        }
        constant Authenticate-Only {
            8;
        }
        constant Authorize-Only {
            17;
        }
        constant Call-Check {
            10;
        }
    }
}

```

```
        constant Callback-Administrative {
            11;
        }
        constant Callback-Framed {
            4;
        }
        constant Callback-Login {
            3;
        }
        constant Callback-NAS-Prompt {
            9;
        }
        constant Fax {
            13;
        }
        constant Framed {
            2;
        }
        constant IAPP-AP-Check {
            16;
        }
        constant IAPP-Register {
            15;
        }
        constant Login {
            1;
        }
        constant Modem-Relay {
            14;
        }
        constant NAS-Prompt {
            7;
        }
        constant Outbound {
            5;
        }
        constant Voice {
            12;
        }
        format integer;
        type 6;
    }
}
attribute Session-Timeout {
    radius {
        format integer;
        type 27;
    }
}
attribute State {
    radius {
        format string;
        type 24;
    }
}
attribute TeliaSonera-Chargeable-User-Id {
    radius {
        format string;
        type 192;
        vendor-id 15297;
    }
}
```

```
}
attribute TeliaSonera-Location-Info {
  radius {
    format string;
    type 194;
    vendor-id 15297;
  }
}
attribute TeliaSonera-Location-Name {
  radius {
    format string;
    type 195;
    vendor-id 15297;
  }
}
attribute TeliaSonera-Operator-Name {
  radius {
    format string;
    type 193;
    vendor-id 15297;
  }
}
attribute TeliaSonera-Visited-Operator-ID {
  radius {
    format string;
    type 196;
    vendor-id 15297;
  }
}
attribute Termination-Action {
  radius {
    constant Default {
      0;
    }
    constant RADIUS-Request {
      1;
    }
    format integer;
    type 29;
  }
}
attribute Tunnel-Assignment-ID {
  radius {
    format string;
    tagged;
    type 82;
  }
}
attribute Tunnel-Client-Auth-ID {
  radius {
    format string;
    tagged;
    type 90;
  }
}
attribute Tunnel-Client-Endpoint {
  radius {
    format string;
    tagged;
    type 66;
  }
}
```

```
}
attribute Tunnel-Medium-Type {
  radius {
    constant 802 {
      6;
    }
    constant ATM {
      3;
    }
    constant Appletalk {
      12;
    }
    constant BBN-1822 {
      5;
    }
    constant Banyan-Vines {
      14;
    }
    constant Decnet-IV {
      13;
    }
    constant E.163 {
      7;
    }
    constant E.164 {
      8;
    }
    constant E.164-NSAP-subaddress {
      15;
    }
    constant F.69 {
      9;
    }
    constant Frame-Relay {
      4;
    }
    constant IP {
      1;
    }
    constant IPX {
      11;
    }
    constant X.121 {
      10;
    }
    constant X.25 {
      2;
    }
    format integer;
    tagged;
    type 65;
  }
}
attribute Tunnel-Password {
  radius {
    format string;
    salt-encrypt;
    tagged;
    type 69;
  }
}
```



```
attribute Tunnel-Preference {
  radius {
    format integer;
    tagged;
    type 83;
  }
}
attribute Tunnel-Private-Group-ID {
  radius {
    format string;
    tagged;
    type 81;
  }
}
attribute Tunnel-Server-Auth-ID {
  radius {
    format string;
    tagged;
    type 91;
  }
}
attribute Tunnel-Server-Endpoint {
  radius {
    format string;
    tagged;
    type 67;
  }
}
attribute Tunnel-Type {
  radius {
    constant AH {
      6;
    }
    constant ATMP {
      4;
    }
    constant DVS {
      11;
    }
    constant ESP {
      9;
    }
    constant GRE {
      10;
    }
    constant IP-IP {
      7;
    }
    constant IP-IP-Tunneling {
      12;
    }
    constant L2F {
      2;
    }
    constant L2TP {
      3;
    }
    constant MIN-IP-IP {
      8;
    }
    constant PPTP {
```

```
        1;
    }
    constant VLAN {
        13;
    }
    constant VTP {
        5;
    }
    format integer;
    tagged;
    type 64;
}
}
attribute User-Name {
    radius {
        format string;
        type 1;
    }
}
attribute User-Password {
    radius {
        format string;
        type 2;
    }
}
}
user@host# show default-model
dictionary radius;

*****

user@host# show shared sic group default-group server
default-server {
    accounting-route {
        test-route {
            editing {
                vpnid;
            }
            target {
                accounting-method default-method;
            }
        }
        default-route {
            target {
                accounting-method default-method;
            }
        }
    }
}
logger {
    debug-logger {
        file {
            filename sic_debug;
            filter /debug-error;
            flush-after-writes;
            maximum-file-size 0;
            prepend-message-header;
            rollover-interval 86400;
        }
        group {
            administration events debug;
            configuration events debug;
            packet events debug;
        }
    }
}
```

```

        packet-trace events debug;
        packet-trace-raw events debug;
        system events debug;
    }
}
default-logger {
    file {
        filename sic;
        filter customized;
        flush-after-writes;
        maximum-file-size 0;
        prepend-message-header;
        rollover-interval 86400;
    }
    group {
        administration events warning;
        configuration events warning;
        packet events debug;
        packet-trace events warning;
        packet-trace-raw events warning;
        system events warning;
    }
}
error-logger {
    file {
        filename sic_error;
        filter /error;
        flush-after-writes;
        maximum-file-size 0;
        prepend-message-header;
        rollover-interval 86400;
    }
    group {
        administration events warning;
        configuration events warning;
        packet events warning;
        packet-trace events warning;
        packet-trace-raw events warning;
        system events warning;
    }
}
}
}

[edit]

```

- Related Topics**
- Overview of Local and Shared Configurations for the SIC (SRC CLI) on page 394
 - Overview of SIC Accounting Methods and Targets (SRC CLI) on page 396
 - SIC Editing Rules Overview (SRC CLI) on page 401
 - Overview of RADIUS Configuration for the SIC (SRC CLI) on page 404

Monitoring the Subscriber Information Collector with the SRC CLI

- Viewing Statistics for Accounting Routes (SRC CLI) on page 461
- Viewing Statistics for RADIUS Client Accounting Requests (SRC CLI) on page 461
- Viewing RADIUS Host Statistics for Accounting Transactions (SRC CLI) on page 461
- Viewing RADIUS Target Statistics for Accounting Requests (SRC CLI) on page 462

Viewing Statistics for Accounting Routes (SRC CLI)

Purpose View accounting route statistics for the SIC, the server collects and displays statistics for each routing rule defined in the SIC server (implicit, explicit, and default).

Action user@host> **show sic statistics route accounting**

Viewing Statistics for RADIUS Client Accounting Requests (SRC CLI)

Purpose View RADIUS client statistics for accounting requests. Statistics are presented for any client from which the server has received packets.

Action user@host> **show sic statistics radius client accounting**

Viewing RADIUS Host Statistics for Accounting Transactions (SRC CLI)

Purpose View RADIUS host statistics for accounting transactions, as well as server runtime and packet error statistics.

Action user@host> **show sic statistics radius host accounting**

```
RADIUS Host Accounting Statistics
Name as accounting server      SIC
Up time:                      6791110
Reset time:                    0
Server status:                 4
Requests:                     1660
Invalid requests:              0
Duplicate requests:            0
Responses:                     1660
Malformed requests:            0
Bad authenticators:            0
Packets dropped:                0
No records:                    0
```

```
Packets of unknown types:      0
Response from invalid addresses: 0
Name as accounting client      SIC
```

Viewing RADIUS Target Statistics for Accounting Requests (SRC CLI)

Purpose View RADIUS target statistics for accounting requests. Statistics are available for RADIUS dynamic authorization and authentication targets that are defined in the server.

Action `user@host> show sic statistics radius target accounting`

PART 9

Index

- Index on page 465

Index

A

- access DNSs.....181
- accounting
 - SAE, description.....9
- ACP (Admission Control Plug-In)
 - redundancy
 - monitoring.....291
- ACP. *See* SRC ACP
- action congestion points.....225
 - configuring260
 - monitoring
 - C-Web interface.....300, 302
 - SRC CLI.....287, 288
- address pools. *See* IP address pools
- Admission Control Plug-In. *See* SRC ACP
- agents *See* NIC agents
- allocating bandwidth to applications not controlled
 - by SRC ACP.....227
- APIs
 - SRC ACP.....231
- APIs (application programming interfaces)
 - CORBA remote API.....8
 - NIC.....175
 - provided with SAE.....7
 - SAE core API.....8
- application programming interfaces. *See* APIs
- applications
 - executing with SRC ACP.....225
 - external for use with SRC ACP.....225, 228
- assigned IP subscribers
 - third-party devices.....93
 - IP address pools.....93
- assigning
 - edge congestion points to subscribers.....257
 - interfaces to backbone congestion point
 - profiles.....267
 - interfaces to subscribers.....257
- ATM access network, using with SRC ACP.....225

- authentication plug-ins

- virtual routers

- SRC CLI.....49, 69

- authorizing and tracking services.....229

B

- backbone congestion point profiles
 - configuring.....267
- backbone congestion points.....241
 - configuring.....260
 - configuring for services.....261
 - defining applications in.....225
 - deriving.....226
 - DNs of.....227
 - monitoring
 - SRC CLI.....286, 287
 - running applications from.....260
- backbone network.....223
- backbone network management with SRC ACP
 - configuring.....259
- background bandwidth.....227
- bandwidth
 - allocating to applications not controlled by SRC
 - ACP.....227
 - background.....227
 - configuring
 - for services.....258, 261
 - for subscribers.....256
 - downstream.....224
 - upstream.....224
- bandwidths and congestion points for subscribers
 - configuring.....256
- basic group
 - configuration
 - SRC CLI.....417, 449
- BEEP, JUNOS routing platforms.....4
 - configuring port
 - SRC CLI.....70
 - connection.....66
- Blocks Extensible Exchange Protocol. *See* BEEP

C

certificate authority (CA).....	72
classification scripts	
congestion point classification	
configuring.....	270
criteria.....	269, 272
description.....	269
how it works.....	269
targets.....	269, 271
Common Object Request Broker Architecture. <i>See</i>	
CORBA	
community manager	
configuring, third-party devices	
SRC CLI.....	100
component interactions	
JUNOS routing platforms and SAE.....	4
configuration group, JUNOS routing	
platforms.....	66, 80
configuration manager, instantiating for NIC.....	176
congestion point applications	
SPI for ACP.....	259
congestion point classification.....	226, 227
congestion point classification scripts. <i>See</i>	
classification scripts	
congestion point expressions.....	227, 276
congestion point profiles.....	226
congestion point expressions.....	276
defining.....	276
congestion points.....	223, 224
configuring.....	256
defining applications in.....	225
deriving.....	225
deriving from congestion point	
expressions.....	227
deriving from profile.....	226
managing.....	228
modifying.....	280
monitoring.....	290
retrieving information about.....	228
conventions	
notice icons.....	xxix
text.....	xxix
COPS (Common Open Policy Service)	
connection with JUNOSe routers.....	45
configuring SAE, SRC CLI.....	50
disabling on router.....	58
enabling on router.....	57
COPS-PR versus COPS XDR.....	4
JUNOSe router connection.....	3

CORBA (Common Object Request Broker

Architecture)	
IOR location.....	172
remote API.....	8
CORBA interfaces	
SRC ACP.....	248
CORBA-based plug-in SPI. <i>See</i> plug-ins, external	
customer support.....	xxxi
contacting JTAC.....	xxxi
customized interface modules.....	8

D

database accounting method	
configuration	
SRC CLI.....	420
deriving congestion points.....	225
device drivers	
JUNOS	
configuring, SRC CLI.....	69
viewing state, C-Web interface.....	85
viewing state, SRC CLI.....	83
viewing statistics, C-Web interface.....	86
viewing statistics, SRC CLI.....	83, 84
JUNOSe	
configuring, SRC CLI.....	50
viewing state, SRC CLI.....	60
viewing statistics, SRC CLI.....	61
directory	
services for SRC ACP.....	252
subscribers for SRC ACP.....	250
directory blacklist, deleting.....	34, 39
distinguished name. <i>See</i> DN	
DN (distinguished name)	
NIC resolution.....	181
DNs	
backbone congestion points.....	227
edge congestion points.....	225
documentation	
comments on.....	xxxi
domain maps	
reloading on SAE.....	39
downstream bandwidth.....	224
downstream RADIUS network elements and	
accounting targets	
configuration	
SRC CLI.....	431

E

edge congestion points	
assigning to subscribers.....	257
deriving.....	225
DNs of.....	225
monitoring	
SRC CLI.....	284, 285
edge network.....	223, 255
edge network management, configuring.....	255
equipment registration	
deleting.....	35, 40
event notification, PCMM network	
configuration statements.....	102
properties, configuring	
SRC CLI.....	102
event notification, third-party devices	
description.....	94
events, publishing.....	241
external applications	
displaying information from.....	289
interaction with NIC.....	175
monitoring	
C-Web interface.....	305, 306
external plug-ins	
configuring SRC ACP as.....	223
external plug-ins. <i>See</i> plug-ins	
External Subscriber Monitor	
acting as pseudo RADIUS server, C-Web	
interface.....	313
agent process statistics, viewing	
SRC CLI.....	328
configuring.....	314
configuring basic local properties.....	314
configuring client secret.....	321
configuring directory connection	
properties.....	315
configuring event notification.....	322
configuring eventing properties.....	316
configuring initial properties.....	315
configuring logging destinations.....	316
event notifications, monitoring	
SRC CLI.....	327
event notifications, viewing	
SRC CLI.....	326
IP address manager.....	313
C-Web Interface.....	313
overview, C-Web interface.....	313
starting.....	323

statistics, monitoring	
SRC CLI.....	326
statistics, viewing	
SRC CLI.....	325
stopping.....	323

F

failover parameters, SAE.....	35, 41
fault recovery, SRC ACP.....	231
files	
ACP data.....	245

G

group	
creation	
SRC CLI.....	417
groups, NIC hosts.....	115

H

hosted internal plug-in.....	241
hosted plug-ins. <i>See</i> plug-ins	

I

interactions between SRC ACP and other	
components.....	228, 230
interface classification scripts	
reloading on SAE.....	33, 39
interface modules, SAE.....	8
interfaces, assigning to backbone congestion point	
profiles.....	267
internal plug-ins. <i>See</i> plug-ins	
IOR	
router initialization scripts.....	53, 104
IP address pools	
local address pools, configuring	
SRC CLI.....	49, 69
static pools, configuring	
SRC CLI.....	49, 69

J

JUNOS routing platforms	
BEEP connection.....	4
configuring port, SRC CLI.....	70
configuration groups.....	66, 80
configuring to interact with SAE.....	78
default virtual router.....	66
disabling interactions with SAE.....	81
enabling interactions with SAE.....	81
monitoring interactions with SAE.....	81

SAE interactions.....	4
SRC software process.....	66
troubleshooting.....	82
JUNOSe routers	
accessing router CLI.....	57
COPS connection.....	3
configuring, SRC CLI.....	50
integration overview.....	45
monitoring interactions with SAE.....	58
router objects, adding	
SRC CLI.....	47
SRC client.....	45
starting.....	57
stopping.....	58
troubleshooting.....	59
VR objects	
adding individually, SRC CLI.....	48, 68
discovering, SRC CLI.....	47

L

LDAP access. <i>See</i> SAE (service activation engine), configuring	
local properties	
configuration	
SRC CLI.....	416
logging properties	
configuring for SRC ACP.....	243
login names.....	181
login process	
assigned IP subscribers, third-party	
devices.....	94
event notification method, third-party	
devices.....	95
login registration	
deleting.....	34, 40

M

managing	
congestion points.....	228
edge network with SRC ACP.....	255
manuals	
comments on.....	xxxi
modifying congestion points.....	280
monitoring	
backbone congestion points.....	241
SRC ACP	
C-Web interface.....	293
SRC CLI.....	283

N

NAS port ID.....	225
network devices	
SNMP communities, configuring	
SRC CLI.....	107
network information collector. <i>See</i> NIC	
network interfaces.....	255, 259
network publisher <i>See</i> NIC	
NIC (network information collector).....	111
API.....	175
configuration prerequisites	
C Series Controllers.....	129
configuration statements.....	127
configuration, changing.....	147
configuration, verifying.....	143
data mapping.....	113
default operating properties, viewing.....	131
factory interface.....	175, 176
logging	
changing configuration.....	136
default.....	136
monitors	
example.....	213
network publisher	
overview.....	149
prerequisites.....	151
running.....	157
troubleshooting.....	159
operating properties, changing.....	131
overview.....	111
planning implementation.....	117
realms	
overview.....	181
replication	
groups.....	115
overview.....	115
SAE plug-in agents.....	141
replication, configuring.....	133
resolution processes.....	181, 182
resolvers	
constraints.....	184
overview.....	114
restarting.....	145
roles.....	182
starting.....	144
stopping.....	144
testing	
any key.....	171
examples.....	144

- resolution.....144
 - test data.....171
 - viewing
 - configuration.....187
 - See also* other NIC entries
 - NIC agents
 - configuration overview.....122
 - directory, configuring
 - SRC CLI.....137
 - overview.....114
 - restarting.....146
 - sae client agents, configuring.....139
 - sae plug-in agents, configuring.....140
 - NIC configuration scenarios
 - changing.....147
 - SRC CLI.....134
 - Multipop.....216
 - OnePop.....188
 - OnePopAcctId.....201
 - OnePopAllRealms.....212
 - OnePopDnSharedIp.....208
 - OnePopDynamicIp.....192
 - OnePopLogin.....203
 - OnePopLoginPull.....205
 - OnePopPcmm.....190
 - OnePopPrimaryUser.....206
 - OnePopSharedIp.....194
 - OnePopStaticRouteIp.....149, 196
 - OnePopVrflp.....149, 199
 - overview.....118, 187
 - scenario-name.....147
 - NIC hosts
 - configuration prerequisites.....130
 - groups.....115
 - overview.....113
 - redundancy
 - example.....212
 - starting.....144
 - stopping.....145, 146
 - NIC locators
 - external applications.....173, 174
 - overview.....111, 113
 - NIC proxies
 - cache, configuring
 - SRC CLI.....168
 - configuration overview.....163
 - configuration prerequisites.....164
 - instantiating.....177
 - logging.....177
 - NIC replication, configuring
 - SRC CLI.....169
 - overview.....113
 - prerequisites.....164
 - removing instances.....180
 - requirements.....175
 - resolution information, configuring
 - SRC CLI.....166
 - resolution requests.....178
 - NIC Proxy for Pseudo-RADIUS server
 - configuring.....318
 - NIC proxy for Pseudo-RADIUS server
 - changing configuration.....318
 - configuring for NIC replication.....319
 - configuring resolution.....318
 - NIC resolvers
 - restarting.....146
 - nic-network-publisher-configuration-statements...150
 - notice icons.....xxix
- O**
- operation
 - SRC ACP, configuring.....244
- P**
- PacketCable Multimedia. *See* PCMM
 - PCMM (PacketCable Multimedia)
 - SAE connection.....4
 - plug-ins
 - architecture.....6
 - authentication
 - virtual routers, SRC CLI.....49, 69
 - external.....6
 - hosted.....7
 - hosted internal plug-in.....241
 - internal.....6
 - SRC ACP.....223
 - tracking
 - virtual routers, SRC CLI.....49, 69
 - types.....5
 - preventing
 - service activation.....229
 - priorityList.....170
 - properties
 - SRC ACP.....243
 - proxy RADIUS accounting method
 - configuration
 - SRC CLI.....421

pseudo-RADIUS server	
configuring External Subscriber Monitor.....	320
publishing events.....	241

R

RADIUS accounting listener	
configuration	
SRC CLI.....	422
randomPick.....	170
realm	
See NIC realms	
redundancy, SRC ACP.....	230
resolution processes	
DN to SAE reference.....	208, 212, 219
IP address to login name.....	203
IP address to SAE	
reference.....	188, 194, 208, 212, 217
login name to SAE reference.....	203
roles, NIC.....	182
roundRobin.....	170
router initialization scripts	
developing.....	53, 103
iorPublisher.....	53, 104
JUNOS	
configuring location, SRC CLI.....	77
JUNOSE	
configuring location, SRC CLI.....	55
example.....	55, 105
JUNOSE Software.....	53, 103
poolPublisher.....	53, 104
specifying for NIC.....	124
router object	
adding for third-party devices	
SRC CLI.....	98
routers	
accessing router CLI.....	57
adding JUNOS routing platforms	
SRC CLI.....	66
adding JUNOSE	
SRC CLI.....	46
integrating JUNOS routing platform.....	65
integrating JUNOSE.....	45
SNMP communities, configuring	
SRC CLI.....	107

S

SAE (service activation engine)	
accounting.....	9
APIs. See APIs	

BEEP connection, JUNOS routing platforms.....	4
COPS	
JUNOSE router connection.....	4
deleting directory blacklist.....	34, 39
disabling interactions with JUNOS routing	
platform.....	81
enabling interactions with JUNOS routing	
platform.....	81
failover parameters.....	35, 41
JUNOS routing platform client.....	78
monitoring interactions	
JUNOS routing platform.....	81
JUNOSE routers.....	58
NIC replication, configuring	
SRC CLI.....	141
overview.....	3
PCMM environment.....	4
plug-ins See plug-ins	
reloading configuration.....	32, 37
role.....	3
router initialization scripts. See router	
initialization scripts	
session store	
C Series Controllers.....	23
starting	
SRC client on JUNOSE router.....	57
stopping	
SRC client on JUNOSE router.....	58
SAE (service activation engine), configuring	
BEEP connection	
SRC CLI.....	70
COPS connection	
SRC CLI.....	50
directory eventing, SAE configuration data	
SRC CLI.....	21
event notification API properties	
SRC CLI.....	102
LDAP access, SRC CLI	
device data.....	20
directory data.....	14
persistent login cache data.....	19
policy data.....	18
service data.....	17
subscriber data.....	15
router initialization script location	
SRC CLI.....	55, 77
serialized data compression	
SRC CLI.....	28

session job manager		
SRC CLI.....	29	
session store		
SRC CLI.....	25	
SRC ACP.....	241	
SAE (service activation engine),configuring		
to monitor backbone congestion points.....	241	
SAE communities		
configuring, third-party devices		
SRC CLI.....	100	
description, third-party devices.....	92	
SAE remote interface		
customized interface modules.....	8	
script services		
for third-party devices.....	92	
serialized data compression, configuring		
SRC CLI.....	28	
servers in group		
configuration		
SRC CLI.....	419	
service activation engine. <i>See</i> SAE		
services		
configuring bandwidth for.....	258, 261	
monitoring		
C-Web interface.....	296	
SRC CLI.....	285	
preventing activation.....	229	
reloading on SAE.....	33, 38	
session job manager, configuring		
SRC CLI.....	29	
session state registrar <i>See</i> SSR		
session store		
C Series Controllers.....	23	
configuring, SRC CLI		
compressing session objects.....	28	
global parameters.....	27	
in third-party networks.....	92	
SIC (subscriber information collector)		
accounting listener		
overview.....	404	
accounting methods		
overview.....	396	
configuring basic SIC group		
.....	449	
database accounting method		
overview.....	396	
overview mapping attributes.....	397	
device models		
overview.....	407	
Device models		
.....	428	
device models, configuration		
SRC CLI.....	429	
dictionaries		
loading.....	408, 428	
merging.....	408	
modifying.....	408	
overview.....	407	
replacing.....	408	
dictionaries, configuration		
SRC CLI.....	426	
downstream network element		
overview.....	404	
editing rules		
overview.....	401	
editing rules, configuration		
SRC CLI.....	439	
editing rules, configuration statements		
SRC CLI.....	435	
event logging		
event level.....	410	
log file.....	410	
log stream.....	410	
overview.....	410	
event logging, configuration		
SRC CLI.....	445	
explicit routing rules		
overview.....	399	
explicit routing, configuration		
SRC CLI.....	423, 425, 442	
explicit routing, configuration statements		
SRC CLI.....	441	
implicit routing rules		
overview.....	401	
implicit routing, configuration		
SRC CLI.....	444	
local and shared configuration		
overview.....	394	
local realms		
overview.....	409	
local realms, configuration		
SRC CLI.....	435	
overview.....	393	
proxy accounting method		
overview.....	398	
proxy function		
overview.....	404	

RADIUS configuration		information from external applications,	
overview.....	404	displaying.....	289
RADIUS transports		interactions with other components.....	228
overview.....	407	logging properties, configuring.....	243
request routing rules		monitoring	
overview.....	399	C-Web interface.....	293
server instance, creation		SRC CLI.....	283
SRC CLI.....	419	operation, configuring.....	244
SNMP support		preventing service activation.....	229
overview.....	413	properties.....	243
snmp, configuring		redundancy.....	230
SRC CLI.....	448	configuring.....	249
statistics		SAE, configuring for	241
accounting routes.....	461	starting.....	279
accounting transactions.....	461	state synchronization.....	230
client accounting requests.....	461	configuring.....	249
target accounting requests.....	462	stopping.....	279
upstream network element		subscribers, monitoring.....	283, 293
overview.....	404	supporting multiple SAEs.....	228
SNMP		using multiple SRC ACPs.....	228
retrieving information from network		SRC ACP (SRC Admission Control Plug-In),	
devices.....	107	congestion points.....	223, 241
SNMP communities		SRC Admission Control Plug-In. <i>See</i> SRC ACP	
configuring		SRC client, JUNOSe routers	
SRC CLI.....	107	configuring.....	45
SRC ACP (SRC Admission Control Plug-In).....	255	starting.....	57
API.....	231	stopping.....	58
ATM access network.....	225	SRC software process, JUNOS routing	
authorizing and tracking services.....	229	platforms.....	66
backbone network management,		disabling.....	81
configuring.....	259	reenabling.....	81
classification scripts		SSR (session state registrar)	
configuring.....	254	adding a client node to an active cluster	
configuring.....	236	SRC CLI.....	375
congestion points.....	223, 241	adding a management server to a client node	
connections to services directory,		in an active cluster	
configuring.....	252	SRC CLI.....	377
connections to subscribers' directory,		adding data nodes to an active cluster	
configuring.....	250	SRC CLI.....	374
CORBA interfaces, configuring.....	248	attribute associations, configuring in an active	
data files.....	245	cluster	
data files, reorganizing.....	280	SRC CLI.....	373
description of.....	223	cluster configurations.....	337
event publishers, configuring.....	241	cluster network requirements.....	339
external applications.....	225, 228	cluster status, viewing	
external plug-in for SAE, configuring.....	223	SRC CLI.....	385
fault recovery.....	231	configuration statements.....	364
groups, configuring.....	236		

- configuring
 - cluster name.....366
 - cluster nodes.....368
 - geometry.....367
 - initial cluster.....365
 - management server.....369
 - subscriber sessions schema.....370, 371
- creating
 - database382
- database mode, viewing
 - SRC CLI.....385
- database modes.....351
- database schema.....347
- database schema, configuring in an active
 - cluster
 - SRC CLI.....372
- database, viewing running configuration
 - SRC CLI.....386
- deleting
 - all subscriber and service sessions.....384
 - database382
 - subscriber sessions by IP address.....384
- disabling
 - SSR component database383
- distributing the cluster configuration.....352
- enabling
 - SSR component database383
- impact of configuration changes.....363
- making modifications to subscriber sessions
 - table.....351
- node groups.....335
- node types.....334
- overview.....333
- placing
 - SSR database into maintenance
 - mode.....381
 - SSR database into production mode.....381
- planning the cluster topology.....355
- planning worksheets.....357
- removing
 - client node.....379
 - data nodes.....378
 - management server.....380
- restarting
 - SSR component database383
- scaling the cluster.....338
- server requirements.....336
- supported configurations.....341
- viewing, all subscriber sessions
 - SRC CLI.....387
- viewing, subscriber sessions by indexed field
 - SRC CLI.....389
- viewing, subscriber sessions by IP address
 - SRC CLI.....388
- viewing, total number of subscriber sessions
 - SRC CLI.....390
- starting
 - SRC ACP.....279
- state synchronization
 - SRC ACP.....230
- statistics, SRC ACP
 - monitoring
 - C-Web interface.....308, 309
- stopping SRC ACP.....279
- subscriber information collector *See* SIC
 - basic group, configuring.....417, 449
 - database accounting method,
 - configuring.....420
 - device models, configuring *See* SIC
 - dictionaries, configuring *See* SIC
 - downstream RADIUS network elements and
 - accounting targets, configuring.....431
 - editing rules, configuration statements *See* SIC
 - editing rules, configuring *See* SIC
 - event logging, configuring *See* SIC
 - explicit routing, configuration statements *See* SIC
 - explicit routing, configuring *See* SIC
 - group, creating.....417
 - implicit routing, configuring *See* SIC
 - local properties
 - directory connection properties,
 - configuring.....416
 - local realms, configuring *See* SIC
 - outbound RADIUS transport for group,
 - configuring *See* SIC
 - proxy RADIUS accounting method,
 - configuring.....421
 - RADIUS accounting listener, configuring.....422
 - RADIUS transport for server, configuring *See* SIC
 - server instance, creating *See* SIC
 - servers in group, configuring.....419
 - snmp, configuring *See* SIC
 - upstream RADIUS network elements and
 - clients.....430

subscribers	
assigning interfaces to.....	257
configuring bandwidths and congestion points for.....	256
IP addresses.....	181
login names.....	181
monitoring	
C-Web interface.....	293, 304
SRC CLI.....	283, 289
provisioned and actual bandwidths.....	228
subscriptions	
reloading on SAE.....	33, 38
support, technical	See technical support

T

targets. See classification scripts	
technical support	
contacting JTAC.....	xxxi
text conventions defined.....	xxix
third-party devices	
creating sessions.....	93
integrating into SRC network	
SRC CLI.....	91
logging in subscribers	
assigned IP method.....	93
overview.....	93
provisioning with script services.....	92
router objects, adding	
SRC CLI.....	98
SAE communities.....	92
VR objects, adding	
SRC CLI.....	99
threads	
configuring for sessions	
SRC CLI.....	29
tracking plug-ins	
virtual routers	
SRC CLI.....	49, 69
troubleshooting	
JUNOS routing platforms.....	82
JUNOSe routers.....	59
tuning factors for background bandwidth.....	227

U

upstream bandwidth.....	224
upstream RADIUS network elements and clients	
configuration	
SRC CLI.....	430

V

virtual routers	
adding for third-party devices	
SRC CLI.....	99
adding individually for JUNOSe routers	
SRC CLI.....	48, 68
adding operative VRs.....	66
SRC CLI.....	47