



SRC-PE Software

Network Guide: SAE, Juniper Networks Routers, NIC, and SRC-ACP

Release 3.0.x

Juniper Networks, Inc.

1194 North Mathilda Avenue
Sunnyvale, California 94089
USA

408-745-2000

www.juniper.net

Part Number: 530-026630-01, Revision 1

This product includes the following software: Fontconfig, X FreeType library, X Render extension headers, and X Render extension library, copyright © 2001, 2003 Keith Packard.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Keith Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Keith Packard makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

KEITH PACKARD DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL KEITH PACKARD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Juniper Networks, the Juniper Networks logo, JUNOS, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. JUNOS is a trademark 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 3.0.x

Copyright © 2008, 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

15 August 2008—Revision 1

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

YEAR 2000 NOTICE

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

SOFTWARE LICENSE

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

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

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

End User License Agreement

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

1. **The Parties.** The parties to this Agreement are Juniper Networks, Inc. and its subsidiaries (collectively "Juniper"), and the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").
2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, and updates and releases of such software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller. "Embedded Software" means Software which Juniper has embedded in the Juniper equipment.
3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:
 - a. Customer shall use the Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.
 - b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius software on multiple computers requires multiple licenses, regardless of whether such computers are physically contained on a single chassis.
 - c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.
 - d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.
 - e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

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

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use the Embedded Software on non-Juniper equipment; (j) use the Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.
5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.
6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

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

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

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

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

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

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

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

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

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

xxiii

Part 1	Operating the SAE	
Chapter 1	Overview of the SAE	3
Chapter 2	Configuring the SAE (SRC CLI)	13
Chapter 3	Managing SAE Data (SRC CLI)	29
Chapter 4	Managing SAE Data (C-Web Interface)	35
Part 2	Using Juniper Networks Routers in the SRC Network	
Chapter 5	Using JUNOS Routers in the SRC Network (SRC CLI)	41
Chapter 6	Using JUNOS Routing Platforms in the SRC Network (SRC CLI)	79
Part 3	Using Network Devices in the SRC Network	
Chapter 7	Integrating Third-Party Network Devices into the SRC Network (SRC CLI)	79
Part 4	Locating Subscriber Management Information	
Chapter 8	Locating Subscriber Information with the NIC	97
Chapter 9	Configuring NIC (SRC CLI)	113
Chapter 10	Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrflp	135
Chapter 11	Configuring Applications to Communicate with an SAE	147
Chapter 12	Configuring SRC Applications to Communicate with an SAE (SRC CLI)	149
Chapter 13	Developing Applications That Use NIC	157
Chapter 14	NIC Resolution Process	165
Chapter 15	NIC Configuration Scenarios	171
Part 5	Providing Admission Control with SRC-ACP	
Chapter 16	Overview of Providing Admission Control with SRC-ACP	207
Chapter 17	Configuring Admission Control (SRC CLI)	217
Chapter 18	Configuring Congestion Point Classification (SRC CLI)	253
Chapter 19	Managing SRC-ACP (SRC CLI)	263
Chapter 20	Monitoring Admission Control (SRC CLI)	265
Chapter 21	Monitoring Admission Control (C-Web Interface)	275

Part 6**Using External Subscriber Monitor**

Chapter 22	Configuring External Subscriber Events with the SRC CLI	293
Chapter 23	Monitoring External Subscriber Events with the SRC CLI	305
Chapter 24	Monitoring External Subscriber Events with the C-Web Interface	309

Part 7**Index**

Index	313
-------	-----

Table of Contents

	About This Guide	xxiii
	SRC Guides and Release Notes	xxiii
	Audience	xxiii
	Documentation Conventions	xxiii
	Related Juniper Networks Documentation	xxv
	Obtaining Documentation	xxvii
	Documentation Feedback	xxvii
	Requesting Technical Support	xxvii
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 JUNOSe Routers and the SAE	3
	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	10
	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	13
	Configuring Access Through LDAPS to Service and Subscriber Data	14

Configuring Access to Subscriber Data	15
Configuring Access to Service Data	16
Configuring Access to Policy Data	18
Configuring Access to the Persistent Login Cache	19
Configuring the Location of Network Device Data	20
Enabling Automatic Discovery of Changes in SAE Configuration Data	21
Setting the Timeout and Number of Events for SAE Directory Eventing	21
Storing Subscriber and Service Session Data	22
Session Store Files	22
Active and Passive Session Stores	23
Standby SAEs	23
Session Store File Rotation	23
Configuring the Session Store Feature	24
Configuring Session Store Parameters for a Device Driver	24
Configuring Global Session Store Parameters	26
Reducing the Size of Objects for the Session Store Feature	27
Configuring the Number of Threads for Sessions	28

Chapter 3

Managing SAE Data (SRC CLI) 29

Commands to Manage SAE Data	29
Reloading the SAE Data	30
Reloading the SAE Configuration	30
Reloading Services	30
Reloading Subscriptions	31
Reloading Interface Classification Scripts	31
Reloading Domain Maps	31
Removing the Directory Blacklist	31
Removing Login Registrations	31
Removing Equipment Registrations	32
Modifying Failover Server Parameters	32
Shutting Down the Device Drivers	33

Chapter 4

Managing SAE Data (C-Web Interface) 35

Reloading the SAE Data (C-Web Interface)	35
Reloading the SAE Configuration	35
Reloading Services	35
Reloading Subscriptions	36
Reloading Interface Classification Scripts	36
Reloading Domain Maps	36
Removing the Directory Blacklist (C-Web Interface)	36
Removing Login Registrations (C-Web Interface)	37
Removing Equipment Registrations (C-Web Interface)	37
Modifying Failover Server Parameters (C-Web Interface)	38
Shutting Down the Device Drivers (C-Web Interface)	38

Part 2**Using Juniper Networks Routers in the SRC Network****Chapter 5****Using JUNOSe Routers in the SRC Network (SRC CLI) 41**

COPS Connection Between JUNOSe Routers and the SAE	41
Highly Available Connections to JUNOSe Routers	42
Adding JUNOSe Routers and Virtual Routers with the CLI	42
Adding Operative JUNOSe Routers and Virtual Routers	42
Adding Routers Individually	43
Adding Virtual Routers Individually	44
Configuring the SAE to Manage JUNOSe Routers with the CLI	45
How SNMP Obtains Information From Routers for the SRC Software	48
Developing Router Initialization Scripts for JUNOSe Routers, JUNOS Routing Platforms, and Network Devices	91
Interface Object Fields	49
Required Methods	92
Example: Router Initialization Script	93
Specifying Router Initialization Scripts on the SAE with the CLI	51
Accessing the Router CLI	52
Starting the SRC Client on a JUNOSe Router	52
Stopping the SRC Client on a JUNOSe Router	53
Monitoring Interactions Between the SAE and the JUNOSe Router	53
Troubleshooting Problems with Managing JUNOSe Routers	53
Viewing the State of JUNOSe Device Drivers (C-Web Interface)	54
Viewing Statistics for Specific JUNOSe Device Drivers (SRC CLI)	55
Viewing Statistics for All JUNOSe Device Drivers (SRC CLI)	55
Viewing the State of JUNOSe Device Drivers (C-Web Interface)	56
Viewing Statistics for All JUNOSe Device Drivers (C-Web Interface)	56

Chapter 6**Using JUNOS Routing Platforms in the SRC Network (SRC CLI) 79**

BEEP Connection Between JUNOS Routing Platforms and the SAE	59
Adding JUNOS Routing Platforms and Virtual Routers	60
Adding Operative JUNOS Routing Platforms	60
Configuring the SAE to Manage JUNOS Routing Platforms	60
Configuring Secure Connections Between the SAE and JUNOS Routing Platforms	62
Adding the Server Certificate on the Routing Platform	63
Creating a Client Certificate for the Router	64
Adding the Client Certificate on the Router	64
Configuring the SAE to Use TLS	64
Configuring TLS on the SAE	64
Checking Changes to the JUNOS Configuration	65
Setting Up Periodic Configuration Checking	66
Using SNMP to Retrieve Information from JUNOSe Routers and JUNOS Routing Platforms (SRC CLI)	66
Specifying Router Initialization Scripts on the SAE	67

Configuring JUNOS Routing Platforms to Interact with the SAE	68
SAE Tracking for LSPs Configured on JUNOS Routing Platforms	69
Overview of SAE Tracking for LSPs Configured on JUNOS Routing Platforms	69
Configuring Event Tracking for JUNOS LSPs (SRC CLI)	69
Configuring the JUNOS Routing Platform to Apply Changes It Receives from the SAE	70
Disabling Interactions Between the SAE and JUNOS Routing Platforms	70
Monitoring Interactions Between the SAE and JUNOS Routing Platforms	71
Troubleshooting Problems Between the SRC Software and JUNOS Device Drivers	71
Troubleshooting Problems with the SRC Software Process	71
Viewing the State of JUNOS Device Drivers (SRC CLI)	72
Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI)	73
Viewing Statistics for All JUNOS Device Drivers (SRC CLI)	73
Viewing the State of JUNOS Device Drivers (C-Web Interface)	74
Viewing Statistics for Specific JUNOS Device Drivers (C-Web Interface)	75
Viewing Statistics for All JUNOS Device Drivers (C-Web Interface)	75

Part 3

Using Network Devices in the SRC Network

Chapter 7

Integrating Third-Party Network Devices into the SRC Network (SRC CLI) 79

Overview of Integrating Network Devices into the SRC Network	79
SAE Communities	79
Storing Session Data	80
Using Script Services to Provision Third-Party Devices	80
Logging In Subscribers and Creating Sessions	81
Assigned IP Subscribers	81
Login Interactions with Assigned IP Subscribers	82
Event Notification from an IP Address Manager	83
Login with Event Notification	83
Configuration Tasks for Integrating Third-Party Network Devices	84
Setting Up Script Services	85
Adding Objects for Network Devices	85
Setting Up SAE Communities	86
Adding Virtual Router Objects	87
Configuring the SAE Community Manager	88
Specifying the Community Manager in the SAE Device Driver	89
Configuring SAE Properties for the Event Notification API with SRC CLI	89
Developing Router Initialization Scripts for JUNOSe Routers, JUNOS Routing Platforms, and Network Devices	91
Interface Object Fields	91
Required Methods	92
Example: Router Initialization Script	93
Copying Initialization Scripts to the C-series Controller	93

Specifying Initialization Scripts on the SAE	93
Using SNMP to Retrieve Information from Network Devices	94
Using the NIC Resolver in Environments that have Third-Party Devices (C-Web Interface)	94

Part 4

Locating Subscriber Management Information

Chapter 8

Locating Subscriber Information with the NIC **97**

Locating Subscriber Management Information	97
NIC Client/Server Mode	98
NIC Local Host Mode	98
Mapping Subscribers to a Managing SAE	99
NIC Proxies and NIC Locators	99
NIC Hosts	99
NIC Agents	100
NIC Resolvers	100
High Availability for NIC	100
High Availability in Existing NIC Configurations	101
NIC Replication	101
Planning a NIC Implementation	103
NIC Configuration Scenarios	103
NIC Agents Used in the NIC Configuration Scenarios	109
Router Initialization Scripts with NIC Configuration Scenarios	111

Chapter 9

Configuring NIC (SRC CLI) **113**

Configuration Statements for the NIC	113
Configuration Statements for NIC Operating Properties	114
Configuration Statements for NIC Scenarios	114
Configuration Statements for NIC Logging	115
Before You Configure the NIC	115
Configuring the NIC (SRC CLI)	116
Starting the NIC (SRC CLI)	117
Reviewing and Changing Operating Properties for NIC (SRC CLI)	117
Reviewing the Default NIC Operating Properties	117
Changing NIC Operating Properties	118
Configuring NIC Replication (SRC CLI)	119
Configuring a NIC Scenario (SRC CLI)	120
Defining the NIC Configuration to Use	120
Configuring Directory Agents	123
Configuring SAE Client Agents	125
Configuring SAE Plug-In Agents	127
Configuring the SAE to Communicate with SAE Plug-In Agents When You Use NIC Replication	128
Configuring Advanced NIC Features	130
Verifying Configuration for the NIC (SRC CLI)	130
Testing a NIC Resolution (SRC CLI)	130

Stopping a NIC Host on a C-series Controller (SRC CLI)	131
Restarting the NIC (SRC CLI)	131
Restarting a NIC Agent (SRC CLI)	132
Restarting a NIC Resolver (SRC CLI)	132
Changing NIC Configurations (SRC CLI)	133

Chapter 10

Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrflp **135**

Overview of the Network Publisher	135
NIC Document That Maps Subscriber IP Addresses to a JUNOS Interface	136
Configuration Statements for the Network Publisher	136
Before You Configure and Run the Network Publisher	137
Configuring the Network Publisher	138
Configuring Local Configuration for the Network Publisher	138
Configuring Connections Between JUNOS Routing Platforms and the Network Publisher	139
Configuring Router Authentication for the Network Publisher	140
Configuring Routing Table Filters for the Network Publisher	141
Configuring the Connection Between the Network Publisher and the Juniper Networks Database	142
Running the Network Publisher	143
Overview of Files to Test Network Publisher	144
Configuring Information to Test the Network Publisher	144
Troubleshooting Network Publisher Operations	145
Reviewing the Information Collected from a JUNOS Routing Platform	146

Chapter 11

Configuring Applications to Communicate with an SAE **147**

Overview of NIC Proxy Configuration	147
Before You Configure a NIC Proxy	147

Chapter 12

Configuring SRC Applications to Communicate with an SAE (SRC CLI) **149**

Configuration Statements for NIC Proxies	149
Configuring Resolution Information for a NIC Proxy (SRC CLI)	150
Changing the Configuration for the NIC Proxy Cache (SRC CLI)	151
Configuring a NIC Proxy for NIC Replication (SRC CLI)	152
Configuring NIC Test Data (SRC CLI)	154

Chapter 13	Developing Applications That Use NIC	157
	External Application Requirements for NIC	157
	External Non-Java Applications That Use NIC	157
	Creating a NIC Locator to Include with a Non-Java Application	158
	External Java Applications That Use NIC	158
	Developing a Java Application to Communicate with a NIC Proxy	159
	Instantiating a Configuration Manager	160
	Passing a Reference to the Configuration Manager to the NIC Factory	160
	Instantiating the NIC Factory Class	160
	Initializing Logging	161
	Instantiating the NIC Proxy	161
	Managing a Resolution Request	162
	Deleting Invalid Results from the NIC Proxy's Cache	163
	Removing the NIC Proxies	164
	Updating Information About Address Pools	164
Chapter 14	NIC Resolution Process	165
	Overview of the NIC Resolution Process	165
	NIC Realms	165
	Key to Value Resolution	166
	NIC Data Types	166
	Constraints as NIC Data Types	169
Chapter 15	NIC Configuration Scenarios	171
	Overview of NIC Configuration Scenarios	171
	OnePop Scenario	172
	Centralized Configuration	172
	Distributed Configuration	173
	Redundancy	173
	OnePopPcmm Scenario	174
	Centralized Configuration	175
	Distributed Configuration	176
	OnePopDynamicIp Scenario	176
	Centralized Configuration	177
	Distributed Configuration	178
	OnePopSharedIp Scenario	178
	Centralized Configuration	179
	Distributed Configuration	180
	OnePopStaticRouteIp	180
	Centralized Configuration	181
	Distributed Configuration	182
	OnePopVrfIp Scenario	182
	Centralized Configuration	183
	Distributed Configuration	184
	OnePopAcctId Scenario	185

OnePopLogin Scenario	186
Centralized Configuration	187
Distributed Configuration	188
OnePopLoginPull Scenario	189
OnePopPrimaryUser	189
Centralized Configuration	190
Distributed Configuration	191
OnePopDnSharedIp Scenario	191
Centralized Configuration	192
Distributed Configuration	193
OnePopAllRealms Scenario	195
MultiPop Scenario	199
IP Realm	200
Shared IP Realm	202
DN Realm	203

Part 5

Providing Admission Control with SRC-ACP

Chapter 16

Overview of Providing Admission Control with SRC-ACP **207**

Overview of SRC-ACP	207
Deriving Congestion Points Automatically	209
Deriving Edge Congestion Points	209
Deriving Congestion Points from a Profile	210
Deriving Backbone Congestion Points	210
Allocating Bandwidth to Applications Not Controlled by SRC-ACP	211
Use of Multiple SRC-ACPs	212
Interactions Between SRC-ACP and Other Components	212
Redundancy	213
Fault Recovery	214
State Synchronization	214
API for ACP	215

Chapter 17

Configuring Admission Control (SRC CLI) **217**

Configuration Statements for SRC-ACP	217
Configuring SRC-ACP	219
Creating Grouped Configurations for SRC-ACP	220
Configuring Local Properties for SRC-ACP	221
Configuring Basic Local Properties for SRC-ACP	221
Configuring Initial Properties for SRC-ACP	222
Configuring Directory Connection Properties for SRC-ACP	223
Configuring Initial Directory Eventing Properties for SRC-ACP	223
Configuring the SAE for SRC-ACP	224
Configuring SRC-ACP as an External Plug-In	224
Configuring Event Publishers	225
Configuring the SAE to Monitor Interfaces for Congestion Points	225

Configuring SRC-ACP Properties	227
Configuring Logging Destinations for SRC-ACP	227
Configuring SRC-ACP Operation	228
Configuring CORBA Interfaces	232
Configuring SRC-ACP Redundancy	233
Configuring Connections to the Subscribers' Directory	234
Configuring Connections to the Services' Directory	236
Configuring SRC-ACP Scripts and Classification	237
Configuring SRC-ACP to Manage the Edge Network	239
Configuring Network Interfaces in the Directory for the Edge Network	239
Configuring Bandwidths for Subscribers	240
Assigning Network Interfaces to Subscribers	241
Configuring Bandwidths for Services in the Edge Network	242
Configuring SRC-ACP to Manage the Backbone Network	242
Configuring Network Interfaces in the Directory for the Backbone Network	243
Extending SRC-ACP Congestion Points for the Backbone Network	243
Configuring Action Congestion Points	244
Configuring Bandwidths for Services in the Backbone Network	245
Configuring Congestion Points for Services in the Backbone Network	245
Plug-In Attributes for Use with Backbone Congestion Point Expressions	246
Using Functions for Backbone Congestion Point Classification Scripts	249
Configuring Congestion Point Profiles in the Directory	250
Assigning Interfaces to Congestion Point Profiles	251

Chapter 18

Configuring Congestion Point Classification (SRC CLI) 253

Overview of Congestion Point Classification	253
Congestion Point Classification Scripts	253
Congestion Point Profiles	254
Configuration Statements for Congestion Point Classification	254
Classifying Congestion Points	254
Configuring Targets and Criteria for Classification Scripts	254
Configuring Classification Scripts Contents for Classification Scripts	255
Configuring Congestion Point Classification Targets	255
Congestion Point Classification Criteria	256
Defining a Congestion Point Profile	259
Congestion Point Expressions	260
Expressions in Templates for Congestion Point Profiles	260
Methods for Use with Scripting Expressions	260
Match Criteria for Congestion Point Classification	261

Chapter 19	Managing SRC-ACP (SRC CLI)	263
	Starting SRC-ACP	263
	Stopping SRC-ACP	263
	Reorganizing the File That Contains ACP Data	263
	Modifying Congestion Points	263
Chapter 20	Monitoring Admission Control (SRC CLI)	265
	Viewing Information About Subscriber Sessions in the Edge Network	265
	Viewing Edge Congestion Point Information by DN	266
	Viewing Edge Congestion Point Information by Subscriber Session	267
	Viewing Information About Services in the Backbone Network	267
	Viewing Backbone Congestion Point Information by DN	268
	Viewing Backbone Congestion Point Information by Service	268
	Viewing Action Congestion Point Information by Service	269
	Viewing Action Congestion Point Information by Congestion Point	270
	Viewing Information About Subscribers Obtained from External Applications	271
	Viewing Congestion Point Information by DN	272
	Viewing Congestion Point Information by Name	272
	Viewing SNMP Information for Devices	273
	Viewing SNMP Information for the Directory	273
	Viewing SNMP Information for SRC-ACP	273
Chapter 21	Monitoring Admission Control (C-Web Interface)	275
	Viewing Information About Subscriber Sessions in the Edge Network (C-Web Interface)	275
	Viewing Information About Edge Congestion Points by DN	276
	Viewing Information About Edge Congestion Points by Subscriber Session	277
	Viewing Information About Services in a Backbone Network (C-Web Interface)	278
	Viewing Information About Congestion Points in a Backbone Network by Expression	280
	Viewing Information About Congestion Points in a Backbone Network by DN	281
	Viewing Information about Action Congestion Points in a Backbone Network by Service	282
	Viewing Information about Action Congestion Points in a Backbone Network by Expression	283
	Viewing Information About Subscribers Obtained from External Applications (C-Web Interface)	285

Viewing Information About Congestion Points from an External Application by DN	286
Viewing Information About Congestion Points from an External Application by Interface Name	286
Viewing Statistics for the SRC-ACP Configuration (C-Web Interface)	287
Viewing General Statistics for SRC-ACP	287
Viewing Statistics for the SRC-ACP Directory	288
Viewing Device Statistics for SRC-ACP	289

Part 6

Using External Subscriber Monitor

Chapter 22

Configuring External Subscriber Events with the SRC CLI 293

Overview of External Subscriber Monitor	293
Configuring External Subscriber Monitor	294
Configuring Basic Local Properties for External Subscriber Monitor	294
Configuring Initial Properties for External Subscriber Monitor	295
Configuring Directory Connection Properties for External Subscriber Monitor	295
Configuring Eventing Properties for External Subscriber Monitor	296
Configuring Logging Destinations for External Subscriber Monitor	296
Configuring the NIC Proxy for the Pseudo-RADIUS Server	297
Configuring Resolution Information for a NIC Proxy	298
Changing the Configuration for the NIC Proxy Cache	298
Configuring a NIC Proxy for NIC Replication	299
Configuring the Pseudo-RADIUS Server for External Subscriber Monitor	300
Configuring the Client Secret for External Subscriber Monitor	301
Configuring Event Notification for External Subscriber Monitor	302
Starting External Subscriber Monitor	302
Stopping External Subscriber Monitor	303

Chapter 23

Monitoring External Subscriber Events with the SRC CLI 305

Viewing Statistics for External Subscriber Monitor	305
Viewing Statistics for External Subscriber Monitor Event Notifications	306
Viewing Statistics for the Agent Process	307

Chapter 24

Monitoring External Subscriber Events with the C-Web Interface 309

Viewing Statistics for External Subscriber Monitor	309
Viewing Statistics for External Subscriber Event Notifications	309
Viewing Statistics for the Agent Process	309

Part 7

Index

Index313

List of Figures

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
Figure 7: SAE Community	80
Figure 8: Login Interactions with Assigned IP Subscribers	82
Figure 9: Login Interactions with Event Notification Application	83
Figure 10: Communication Between a NIC Proxy and a NIC Host in Client/Server Mode	98
Figure 11: Communication Between a NIC Host and a NIC Proxy in Local Host Mode	99
Figure 12: NIC Groups	101
Figure 13: NIC Group Selection by Round-Robin	102
Figure 14: NIC Resolution Request	103
Figure 15: Resolution Process for ip Realm	172
Figure 16: OnePop Centralized Configuration	173
Figure 17: OnePop Distributed Configuration	173
Figure 18: Redundancy for OnePop Centralized Configuration	174
Figure 19: Resolution Process for Pcomm_am Realm	175
Figure 20: OnePopPcomm Centralized Configuration	176
Figure 21: OnePopPcomm Distributed Configuration	176
Figure 22: Resolution Process for dynamicIp Realm	177
Figure 23: OnePopDynamicIp Centralized Configuration	178
Figure 24: OnePopDynamicIp Distributed Configuration	178
Figure 25: Resolution Process for sharedIp Realm	179
Figure 26: OnePopSharedIP Centralized Configuration	180
Figure 27: OnePopSharedIP Distributed Configuration	180
Figure 28: Resolution Process for the StaticRouteIp Realm	181
Figure 29: OnePopStaticRouteIp Centralized Configuration	182
Figure 30: OnePopStaticRouteIp Distributed Configuration	182
Figure 31: Resolution Process for the VrfIp Realm	183
Figure 32: OnePopVrfIp Centralized Configuration	184
Figure 33: OnePopStaticRouteIp Distributed Configuration	184
Figure 34: Resolution Process for acctId Realm	185
Figure 35: OnePopAcctId Centralized Configuration	186
Figure 36: Resolution Processes login Realm	187
Figure 37: OnePopLogin Centralized Configuration	188
Figure 38: OnePopLogin Distributed Configuration	189
Figure 39: OnePopLoginPull Distributed Configuration	189
Figure 40: Resolution Processes for primary_user Realm	190

Figure 41: OnePopPrimaryUser Centralized Configuration	191
Figure 42: OnePopPrimaryUser Distributed Configuration	191
Figure 43: OnePopDnSharedIp Realms Centralized Configuration	193
Figure 44: OnePopDnSharedIp Realms Distributed Configuration	195
Figure 45: OnePopAllRealms Centralized Configuration	197
Figure 46: OnePopAllRealms Distributed Configuration	199
Figure 47: MultiPop Configuration	200
Figure 48: iP Realm for MultiPop Configuration	201
Figure 49: sharedIP Realm for MultiPop Configuration	203
Figure 50: Resolution Graph for MultiPOP dn Realm	203
Figure 51: dn Realm for MultiPop Configuration	204
Figure 52: Position of SRC-ACP in Network	208

List of Tables

Table 1: Notice Icons	xxiv
Table 2: Text Conventions	xxiv
Table 3: Juniper Networks C-series and SRC Technical Publications	xxv
Table 4: Router Initialization Scripts	91
Table 5: Exported Fields	92
Table 6: Router Initialization Scripts	91
Table 7: Exported Fields	92
Table 8: Types of NIC Agents	100
Table 9: NIC Configuration Scenarios	104
Table 10: NIC Agents	109
Table 11: Agents in Configuration Scenarios	110
Table 12: Type of Router Initialization Script to Use for NIC Configuration Scenarios	111
Table 13: Available NIC Resolutions	165
Table 14: Congestion Points Derived Through NAS Port ID	209
Table 15: Output Fields for show external-subscriber-monitor statistisc radius-accounting	305
Table 16: Output Fields for show external-subscriber-monitor statistics event-notifications	306
Table 17: Output Fields for show external-subscriber-monitor statistics process	307

About This Guide

- SRC Guides and Release Notes on page xxiii
- Audience on page xxiii
- Documentation Conventions on page xxiii
- Related Juniper Networks Documentation on page xxv
- Obtaining Documentation on page xxvii
- Documentation Feedback on page xxvii
- Requesting Technical Support on page xxvii

SRC Guides and Release Notes

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

Audience

This guide is intended for experienced system and network specialists working with JUNOS routers and JUNOS routing platforms in an Internet access environment. We assume that readers know how to use the routing platforms, directories, and RADIUS servers that they will deploy in their SRC networks.

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 xxiv defines the notice icons used in this guide. Table 2 on page xxiv 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.	<code>user@host# set cache-entry-age cache-entry-age</code>
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"> ■ <code>system ldap server{ stand-alone;</code> ■ Use the <code>request sae modify device failover</code> command with the <code>force</code> option ■ <code>user@host# . . .</code> ■ <code>http://www.juniper.net/techpubs/software/management/src/api-index.html</code>
<i>Italic sans serif typeface</i>	Represents variables in SRC CLI commands.	<code>user@host# set local-address local-address</code>
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <code>< gfwif ></code> .
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.srmt.sae.plugin\RADIUSTrackingPluginEvent
Words separated by the symbol	Represent a choice to select one keyword or variable to the left or right of this symbol. (The keyword or variable may be either optional or required.)	diagnostic line

Related Juniper Networks Documentation

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

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

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

Document	Description
Core Documentation Set	
<i>C2000 and C4000 Hardware Guide</i>	Describes the hardware platforms and how to install, maintain, replace, and troubleshoot them. The guide also includes specifications.
<i>C2000 and C4000 Quick Start Guide</i>	Describes how to get the C-series Controller up and running quickly. Intended for experienced installers who want to expedite the installation process.
<i>SRC-PE Getting Started Guide</i>	Describes the SRC software, how to set up an initial software configuration, how to integrate RADIUS servers, and how to upgrade the SRC software. It also explains how to manage a C-series Controller. The guide describes how to set up and start the SRC CLI and the C-Web interface, as well as other SRC configuration tools. It includes reference material for the SRC documentation.
<i>SRC-PE CLI User Guide</i>	Describes how to use the SRC CLI, configure and monitor the platform with the CLI, and control the CLI environment. The guide also describes how to manage SRC components with the CLI.

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

Document	Description
<i>SRC-PE Network Guide: SAE, Juniper Networks Routers, NIC, and SRC-ACP</i>	Describes how to use and configure the SAE, the NIC, and the SRC-ACP (Admission Control Plug-In) application. This guide also provides detailed information about using JUNOSe routers, JUNOS routing platforms, and other network devices in the SRC network.
<i>SRC-PE Services and Policies Guide</i>	Describes how to work with services and policies. The guide provides an overview, configuration procedures, and management information. The guide also provides information about the SRC tools for configuring policies.
<i>SRC-PE Subscribers and Subscriptions Guide</i>	Describes how to work with residential and enterprise subscribers and subscriptions. The guide provides an overview, configuration procedures, and management information. This guide also provides information about the enterprise service portals, including the Enterprise Manager Portal.
<i>SRC-PE Monitoring and Troubleshooting Guide</i>	Describes how to use logging, the SNMP agent, the SRC CLI, and the C-Web interface to monitor and troubleshoot SRC components. This guide also describes the SNMP traps.
<i>SRC-PE Solutions Guide</i>	Provides high-level instructions for SRC implementations. The guide documents the following scenarios: managing QoS services on JUNOSe routers; managing subscribers in a wireless roaming environment; providing voice over IP (VoIP) services; integrating the SRC software in a PCMM environment, including the use of the Juniper Policy Server (JPS); and mirroring subscriber traffic on JUNOSe routers.
<i>SRC-PE CLI Command Reference, Volume 1</i> <i>SRC-PE CLI Command Reference, Volume 2</i>	Together constitute information about command and statement syntax; descriptions of commands, configuration statements, and options; editing level of statement options; and a history of when a command was added to the documentation.
<i>SRC-PE NETCONF API Guide</i>	Describes how to use the NETCONF application programming interface (API) to configure or request information from the NETCONF server on a C-series Controller that runs the SRC software.
<i>SRC-PE XML API Configuration Reference</i>	Describes the tag elements in the SRC Extensible Markup Language (XML) application programming interface (API) that are equivalent to configuration statements in the SRC command-line interface (SRC CLI).
<i>SRC-PE XML API Operational Reference</i>	Describes the tag elements in the SRC Extensible Markup Language (XML) application programming interface (API) that are equivalent to operational commands in the SRC command-line interface (SRC CLI).
Application Library	
<i>SRC Application Library Guide</i>	Describes how to install and work with applications that you can use to extend the capabilities of the SRC software. The guide documents the following applications: SRC-SG (SOAP Gateway) Web applications, an application to provide threat mitigation, an application to provide tracking and QoS control at the application level by integrating the SRC software with the Ellacoya deep packet inspection (DPI) platform, and an application to control volume usage .
Release Notes	

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

Document	Description
<i>SRC-PE Release Notes</i>	In the <i>Release Notes</i> , you will find the latest information about features, changes, known problems, resolved problems, supported platforms and network devices (such as Juniper Networks routers and CMTS devices), and third-party software. If the information in the <i>Release Notes</i> differs from the information found in the documentation set, follow the <i>Release Notes</i> .
<i>SRC Application Library Release Notes</i>	
Release notes are available on the Web.	

Obtaining Documentation

To obtain the most current version of all Juniper Networks technical documentation, see the products documentation page on the Juniper Networks Web site at <http://www.juniper.net/>.

To order a documentation CD, which contains this guide, contact your sales representative.

Copies of the Management Information Bases (MIBs) available in a software release are included on the documentation CDs and at <http://www.juniper.net/>.

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 <http://www.juniper.net/techpubs/docbug/docbugreport.html>. If you are using e-mail, be sure to include the following information with your comments:

- Document name
- Document part number
- Page number
- Software release version (not required for *Network Operations Guides [NOGs]*)

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/customers/support/downloads/710059.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 Manager: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool located at <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 Manager 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).

Part 1

Operating the SAE

- Overview of the SAE on page 3
- Configuring the SAE (SRC CLI) on page 13
- Managing SAE Data (SRC CLI) on page 29
- Managing SAE Data (C-Web Interface) on page 35

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.

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 JUNOS router supports. When you set up your JUNOS 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 JUNOS 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-I03-051221 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 software 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.

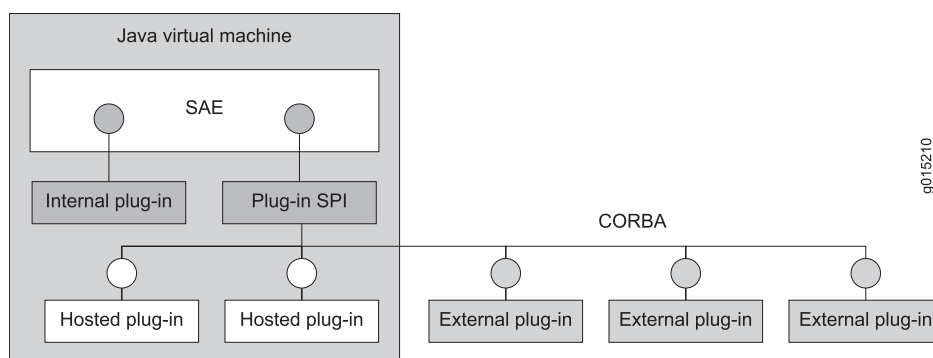
For more information, see .

- Related Topics**
- Adding JUNOSe Routers and Virtual Routers with the CLI
 - Configuring the SAE to Manage JUNOS Routing Platforms
 - Overview of a PCMM Environment
 - Overview of the JPS

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 software. Figure 1 on page 6 gives an overview of the plug-in architecture.

Figure 1: SAE Plug-In Architecture

Internal Plug-Ins

The SRC software 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 software provides a set of internal plug-ins. .

External Plug-Ins

The SRC software 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 software.

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

- Configuring the SAE for External Plug-Ins
- How Internal Plug-Ins Work
- 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>.

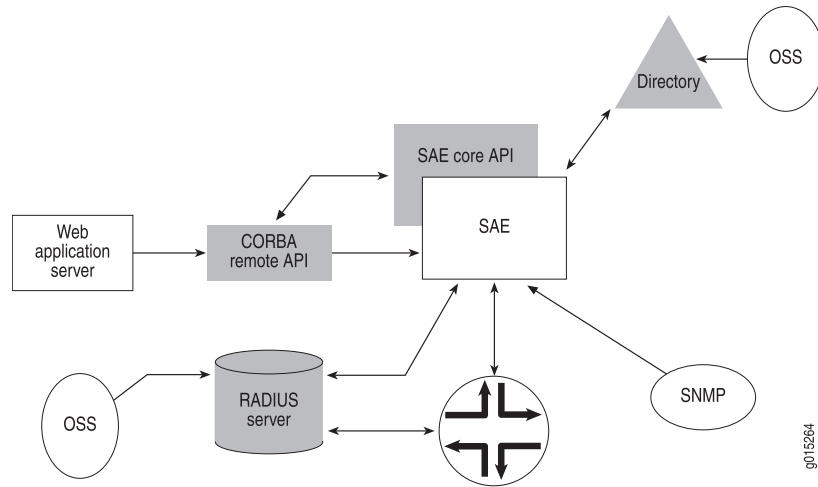
Tracking and Controlling Subscriber and Service Sessions with SAE APIs

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 software. 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

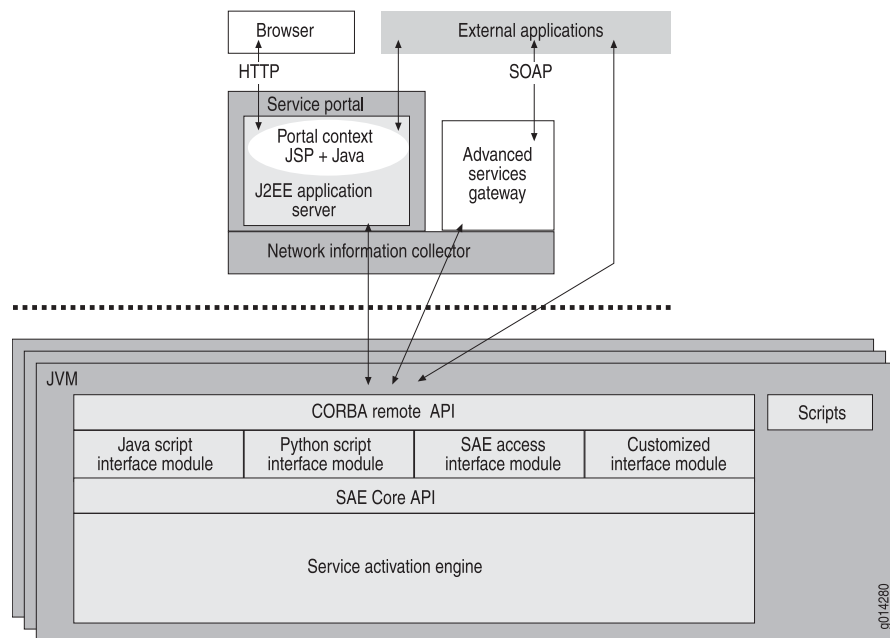
This API provides a way to use external applications with the SRC software (see Figure 3 on page 9). All functions that are available through the SAE core API are available through the CORBA remote API. The remote API provides several remote interfaces that allow customization of the API for special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

- 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>.

SAE Accounting

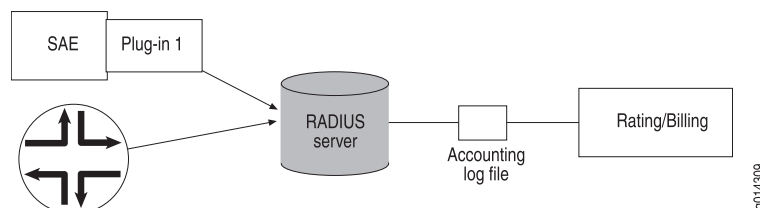
The router and the SAE generate RADIUS accounting records when subscribers access the Internet and use value-added services. The records are sent to RADIUS accounting servers and are logged in accounting log files, or they are sent to accounting flat files. External systems collect the accounting log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

The SRC software 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 software works with other AAA RADIUS servers; however, we validate the SRC software 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

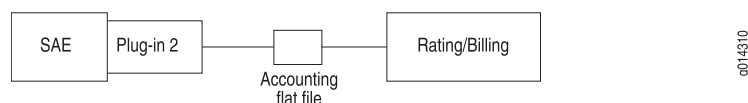
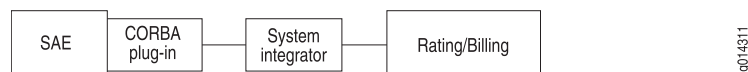


Figure 6 on page 10 illustrates a third possibility, one in which the customer uses a CORBA plug-in of his or her own choice.

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 software 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.

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.

Chapter 2

Configuring the SAE (SRC CLI)

- SRC Access to Directory Data on page 13
- Configuring LDAP Access to Directory Data on page 13
- Configuring Access Through LDAPS to Service and Subscriber Data on page 14
- Configuring Access to Subscriber Data on page 15
- Configuring Access to Service Data on page 16
- Configuring Access to Policy Data on page 18
- Configuring Access to the Persistent Login Cache on page 19
- Configuring the Location of Network Device Data on page 20
- Enabling Automatic Discovery of Changes in SAE Configuration Data on page 21
- Setting the Timeout and Number of Events for SAE Directory Eventing on page 21
- Storing Subscriber and Service Session Data on page 22
- Configuring the Session Store Feature on page 24
- Configuring the Number of Threads for Sessions on page 28

SRC Access to Directory Data

The SRC software 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.

Configuring LDAP Access to Directory Data

The tasks to configure LDAP access to directory data are:

- (Optional) Configuring Access Through LDAPS to Service and Subscriber Data
- Configuring Access to Subscriber Data
- Configuring Access to Service Data
- Configuring Access to Policy Data

- Configuring Access to the Persistent Login Cache
- Configuring the Location of Network Device Data
- Enabling Automatic Discovery of Changes in SAE Configuration Data
- Setting the Timeout and Number of Events for SAE Directory Eventing

Configuring Access Through LDAPS to Service and Subscriber Data

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/IorPublisher.py* script provide examples of how to configure a directory context. For example, from the */opt/UMC/sae/lib/IorPublisher.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
 - Configuring Access to Service Data

Configuring Access to Subscriber Data

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

Configuring Access to Service Data

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 Through LDAPS to Service and Subscriber Data

Configuring Access to Policy Data

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 the Persistent Login Cache

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 the Location of Network Device Data

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

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)

Setting the Timeout and Number of Events for SAE Directory Eventing

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)F

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

In a community of SAEs, one SAE can provide redundancy for the active SAE. The redundant (standby) SAE connects to the active SAE through a COPS-PR connection. State as well as session data is replicated from the active SAE to the standby SAE to reduce the failover time from one SAE to another.

A standby SAE can respond to SAE failures and connection failures between an SAE and a JUNOS router. Connection failures between an active SAE and a standby SAE may not be immediately detected, because each SAE continues to function for a period of time. When a standby SAE does detect that state information is different on the two SAEs, it resynchronizes data between the two.



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.

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.

Configuring the Session Store Feature

You can configure three things for the session store feature:

1. Configuring Session Store Parameters for a Device Driver on page 24
2. Configuring Global Session Store Parameters on page 26
3. Reducing the Size of Objects for the Session Store Feature on page 27

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](#).

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

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;
```


Chapter 3

Managing SAE Data (SRC CLI)

- Commands to Manage SAE Data on page 29
- Reloading the SAE Data on page 30
- Reloading the SAE Configuration on page 30
- Reloading Services on page 30
- Reloading Subscriptions on page 31
- Reloading Interface Classification Scripts on page 31
- Reloading Domain Maps on page 31
- Removing the Directory Blacklist on page 31
- Removing Login Registrations on page 31
- Removing Equipment Registrations on page 32
- Modifying Failover Server Parameters on page 32
- Shutting Down the Device Drivers on page 33

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*.

Reloading the SAE Data

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 with the CLI
- Viewing Information About SAE Device Drivers with the CLI
- Viewing Information About Services with the CLI
- Viewing Information About Policies on the SAE with the CLI

Reloading the SAE Configuration

To reload the SAE configuration data from the directory:

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

The new configuration takes effect immediately.

Related Topics

- Reloading the SAE Data
- Reloading Services

Reloading Services

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 Data

Reloading the SAE Configuration

Reloading Subscriptions

To reload all subscriptions from the directory:

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

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

Reloading Interface Classification Scripts

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
```

Reloading Domain Maps

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.

Removing the Directory Blacklist

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.

Removing Login Registrations

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

```

Removing Equipment Registrations

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

```

Modifying Failover Server Parameters

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
```

Shutting Down the Device Drivers

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
```


Chapter 4

Managing SAE Data (C-Web Interface)

- Reloading the SAE Data (C-Web Interface) on page 35
- Removing the Directory Blacklist (C-Web Interface) on page 36
- Removing Login Registrations (C-Web Interface) on page 37
- Removing Equipment Registrations (C-Web Interface) on page 37
- Modifying Failover Server Parameters (C-Web Interface) on page 38
- Shutting Down the Device Drivers (C-Web Interface) on page 38

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

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.

Reloading Services

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.

Reloading Subscriptions

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.

Reloading Interface Classification Scripts

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**.

Reloading Domain Maps

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.

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**.

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**.

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**.

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**.

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**.

Part 2

Using Juniper Networks Routers in the SRC Network

- Using JUNOSe Routers in the SRC Network (SRC CLI) on page 41
- Using JUNOS Routing Platforms in the SRC Network (SRC CLI) on page 79

Chapter 5

Using JUNOSe Routers in the SRC Network (SRC CLI)

- COPS Connection Between JUNOSe Routers and the SAE on page 41
- Adding JUNOSe Routers and Virtual Routers with the CLI on page 42
- Configuring the SAE to Manage JUNOSe Routers with the CLI on page 45
- How SNMP Obtains Information From Routers for the SRC Software on page 48
- Developing Router Initialization Scripts for JUNOSe Routers, JUNOS Routing Platforms, and Network Devices on page 91
- Specifying Router Initialization Scripts on the SAE with the CLI on page 51
- Accessing the Router CLI on page 52
- Starting the SRC Client on a JUNOSe Router on page 52
- Stopping the SRC Client on a JUNOSe Router on page 53
- Monitoring Interactions Between the SAE and the JUNOSe Router on page 53
- Troubleshooting Problems with Managing JUNOSe Routers on page 53
- Viewing the State of JUNOSe Device Drivers (C-Web Interface) on page 54
- Viewing Statistics for Specific JUNOSe Device Drivers (SRC CLI) on page 55
- Viewing Statistics for All JUNOSe Device Drivers (SRC CLI) on page 55
- Viewing the State of JUNOSe Device Drivers (C-Web Interface) on page 56
- Viewing Statistics for All JUNOSe Device Drivers (C-Web Interface) on page 56

COPS Connection Between JUNOSe Routers and the SAE

Configuring the SRC client on a JUNOSe 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 JUNOSe router. Subsequently, the SRC client sends configuration changes made on the JUNOSe 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

Adding JUNOSe Routers and Virtual Routers with the 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 42
2. Adding Routers Individually on page 43
3. Adding Virtual Routers Individually on page 44

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

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

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 ...];
}
```

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 JUNOSe 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 JUNOSe 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# tracking-plugin [ tracking-plugin ...]
```

9. (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;
```

Configuring the SAE to Manage JUNOSe Routers with the CLI

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

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

```
shared sae configuration driver junose {
  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 JUNOSe routers:

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

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

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

```
[edit shared sae group west-region configuration driver junose]
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 junose]
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)

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 JUNOS 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. See *Adding JUNOS Routers and Virtual Routers with the CLI*.

Developing Router Initialization Scripts for JUNOS Routers, JUNOS Routing Platforms, and Network Devices

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 4 on page 91 describes the router initialization scripts that we provide with the SRC software in the */opt/UMC/sae/lib* folder.

Table 4: Router Initialization Scripts

Script Name	Function	When to Use Script
lorPublisher	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.

Table 4: Router Initialization Scripts *(continued)*

Script Name	Function	When to Use Script
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 JUNOSe 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 5 on page 92 describes the fields that the SAE exports.

Table 5: 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
```

Specifying Router Initialization Scripts on the SAE with the 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 software 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;
```

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
```

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
```


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.

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 ssrc enable
```

Monitoring Interactions Between the SAE and the JUNOSe Router

Purpose Monitor connection between the SAE and a JUNOSe router.

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

- Use the `show ssrc info` command on the JUNOSe router

To display the version number of the SRC client:

- Use the `show ssrc 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 software 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*.

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

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

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 with the JUNOSe router 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.

Viewing Statistics for Specific JUNOSe Device Drivers (SRC CLI)

Purpose Display statistics for a specific JUNOSe 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                      JUNOSe COPS
Up time                          172286
Using failover server             false
```

Viewing Statistics for All JUNOSe Device Drivers (SRC CLI)

Purpose Display SNMP statistics for all JUNOSe 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                   0:0:0:0:0:0:0:0
Server port                      3288
Time since last redirect         186703
```

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

Problem The log files indicate a problem with a specific driver.

Solution Review the configuration of the associated with the JUNOS router driver with the C-Web interface:

1. Select **SAE** from the side pane, and click **Drivers**.

The Drivers pane appears.

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:

< virtual router name > @ < 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.

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

Purpose To view SNMP statistics for all JUNOS 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 JUNOS device driver.

Chapter 6

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

- BEEP Connection Between JUNOS Routing Platforms and the SAE on page 59
- Adding JUNOS Routing Platforms and Virtual Routers on page 60
- Configuring the SAE to Manage JUNOS Routing Platforms on page 60
- Configuring Secure Connections Between the SAE and JUNOS Routing Platforms on page 62
- Adding the Server Certificate on the Routing Platform on page 63
- Creating a Client Certificate for the Router on page 64
- Adding the Client Certificate on the Router on page 64
- Configuring the SAE to Use TLS on page 64
- Configuring TLS on the SAE on page 64
- Checking Changes to the JUNOS Configuration on page 65
- Using SNMP to Retrieve Information from JUNOS Routers and JUNOS Routing Platforms (SRC CLI) on page 66
- Specifying Router Initialization Scripts on the SAE on page 67
- Configuring JUNOS Routing Platforms to Interact with the SAE on page 68
- SAE Tracking for LSPs Configured on JUNOS Routing Platforms on page 69
- Configuring the JUNOS Routing Platform to Apply Changes It Receives from the SAE on page 70
- Disabling Interactions Between the SAE and JUNOS Routing Platforms on page 70
- Monitoring Interactions Between the SAE and JUNOS Routing Platforms on page 71
- Troubleshooting Problems Between the SRC Software and JUNOS Device Drivers on page 71

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.

Adding JUNOS Routing Platforms and Virtual Routers

On JUNOS routing platforms, the SAE manages interfaces. The SRC software 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:

1. Adding Operative JUNOS Routing Platforms on page 60

Adding Operative JUNOS Routing Platforms

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.

Configuring the SAE to Manage JUNOS Routing Platforms

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 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
2. Creating a Client Certificate for the Router
3. Adding the Client Certificate on the Router
4. Configuring the SAE to Use TLS
5. Configuring TLS on the SAE

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;
          }
        }
      }
    }
  }
}
```

```

    }
  }
}

```

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*.

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;
    }
  }
}

```

Configuring the SAE to Use TLS

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* .

Configuring TLS on the SAE

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;
```

Checking Changes to the JUNOS Configuration

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.

Setting Up Periodic Configuration Checking

Use the following configuration statements to configure the SAE to periodically check the configuration of the JUNOS routing platform:

```
shared sae configuration driver junos configuration-checking
configuration-checking-schedule configuration-checking-schedule ;
configuration-checking-action (enforce | synchronize | detect);
```

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;
```

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 with the CLI.) 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 *****;
```

Specifying Router Initialization Scripts on the SAE

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 software 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;
```

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.

SAE Tracking for LSPs Configured on JUNOS Routing Platforms

- Overview of SAE Tracking for LSPs Configured on JUNOS Routing Platforms on page 69
- Configuring Event Tracking for JUNOS LSPs (SRC CLI) on page 69

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

- Configuring Event Tracking for JUNOS LSPs (SRC CLI)

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.



NOTE: Configure LSP tracking at the expert editing level.

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

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;
```

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.

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

Troubleshooting Problems Between the SRC Software and JUNOS Device Drivers

- Troubleshooting Problems with the SRC Software Process on page 71
- Viewing the State of JUNOS Device Drivers (SRC CLI) on page 72
- Viewing Statistics for Specific JUNOS Device Drivers (SRC CLI) on page 73
- Viewing Statistics for All JUNOS Device Drivers (SRC CLI) on page 73
- Viewing the State of JUNOS Device Drivers (C-Web Interface) on page 74
- Viewing Statistics for Specific JUNOS Device Drivers (C-Web Interface) on page 75
- Viewing Statistics for All JUNOS Device Drivers (C-Web Interface) on page 75

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*.
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
```

2. Restart the SRC software process on the router.

```
root@ui1 > restart service-deployment
```

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@troll

```

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

```

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
```

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 ACP, CLI, Component, Date, Disk, Interfaces..., JPS, NIC, NTP, Redirect Server, Route..., SAE (highlighted), Security, and System. The top of the main area says 'Monitor' and 'Logged in as: admin'. Below this, the 'SAE' section is active, showing a 'Drivers' pane. This pane contains three input fields: 'Name Of Device Driver' (with a text box), 'Style' (with a dropdown menu), and 'Maximum Results' (with a text box). To the right of these fields are descriptive text boxes: '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 left of the form area are 'OK' and 'Reset' buttons. The footer of the page contains copyright information and the Juniper logo.

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.

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. On the left is a sidebar with a 'Monitor' header and a list of components: ACP, CLI, Component, Date, Disk, Interfaces..., JPS, MIC, NTP, Redirect Server, Route..., SAE (highlighted), Security, and System. The main content area is titled 'SAE' and 'Device'. It contains a 'Device Name' text input field, a 'Style' dropdown menu, and 'OK' and 'Reset' buttons. To the right of the input fields is a help text box that reads: '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. Output style Choices: brief: Display only device names'. At the top right of the interface, it says 'Logged in as: admin' and has links for 'About', 'Refresh', and 'Logout'. The bottom of the interface shows a copyright notice for Juniper Networks, Inc. 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.

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.

Monitor Logged in as: admin About Refresh Logout

SAE > Statistics > Device > Common

SAE

Common

Device Name

Type

OK Reset

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 JUNOSe router drivers
packetcable-cops: Display SNMP statistics for PCMM device drivers
proxy: Display SNMP statistics for third-party drivers

Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy. Juniper Your Net.

- 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.
- Select the **junos** JUNOS from the Type list:
- Click **OK**.

The Common pane displays statistics for the specified device.

Part 3

Using Network Devices in the SRC Network

- Integrating Third-Party Network Devices into the SRC Network (SRC CLI) on page 79

Chapter 7

Integrating Third-Party Network Devices into the SRC Network (SRC CLI)

- Overview of Integrating Network Devices into the SRC Network on page 79
- Logging In Subscribers and Creating Sessions on page 81
- Configuration Tasks for Integrating Third-Party Network Devices on page 84
- Setting Up Script Services on page 85
- Adding Objects for Network Devices on page 85
- Setting Up SAE Communities on page 86
- Configuring SAE Properties for the Event Notification API with SRC CLI on page 89
- Developing Router Initialization Scripts for JUNOSe Routers, JUNOS Routing Platforms, and Network Devices on page 91
- Copying Initialization Scripts to the C-series Controller on page 93
- Specifying Initialization Scripts on the SAE on page 93
- Using SNMP to Retrieve Information from Network Devices on page 94
- Using the NIC Resolver in Environments that have Third-Party Devices (C-Web Interface) on page 94

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.

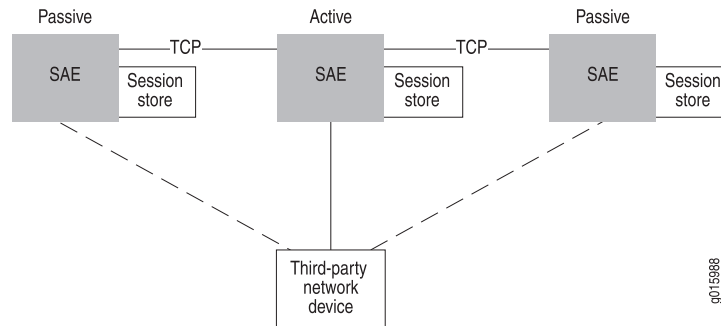
To activate services and provision policies on the device, you use script services. You can also activate aggregate services for subscribers. However, you cannot activate normal services that require policies to be provisioned on the device.

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 80 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*.

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**
- Configuration Tasks for Integrating Third-Party Network Devices
 - Logging In Subscribers and Creating Sessions

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 SOAP Gateway (SRC-SG).

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.

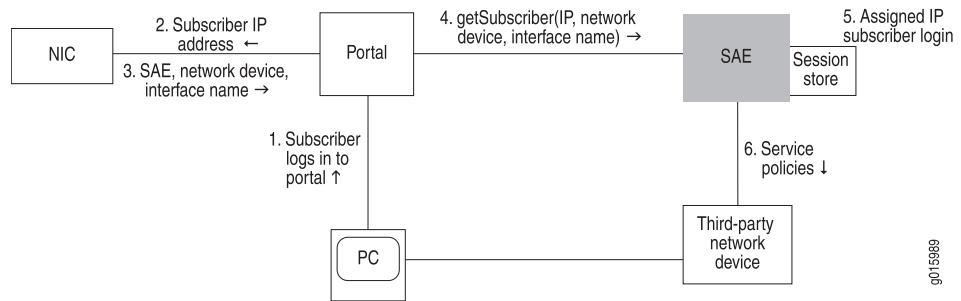
Assigned IP Subscribers

With the assigned IP subscriber method of logging in subscribers and creating sessions, the SRC software 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-SG 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 82, the subscriber activates a service through a portal. You could also have the subscriber activate a service through the SRC-SG.

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 `getSubscriber` 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

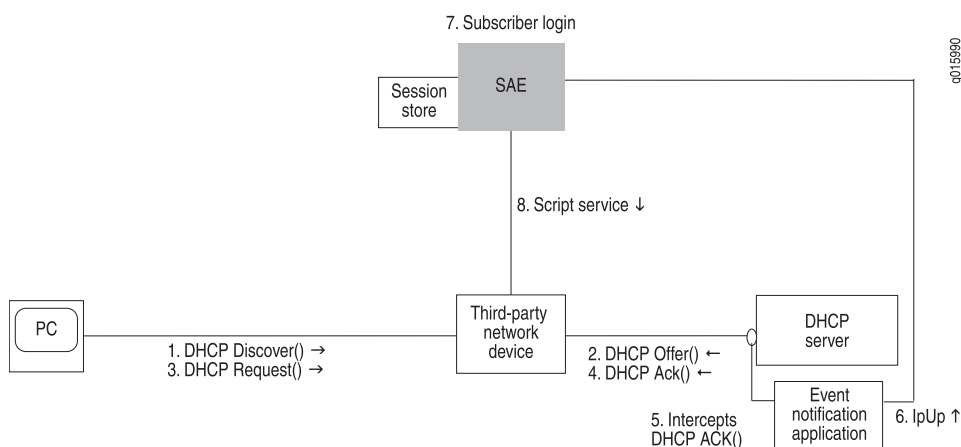
With the event notification method of logging in subscribers and creating subscriber sessions, the subscriber logs in to the network device and obtains an IP address through an address server, usually a DHCP server. The SAE receives notifications about the subscriber, such as the subscriber's IP address, 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
 - Configuration Tasks for Integrating Third-Party Network Devices

Configuration Tasks for Integrating Third-Party Network Devices

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](#) .
- Add objects for the devices.
See [Adding Objects for Network Devices](#) .
- Configure an SAE community.
See [Setting Up SAE Communities](#) .
- (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 with SRC CLI](#).
- Configure the session store.
See [Storing Subscriber and Service Session Data](#).
- 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*.

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
 - Overview of Integrating Network Devices into the SRC Network
 - Logging In Subscribers and Creating Sessions

Adding Objects for Network Devices

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 configuration 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**
- Configuration Tasks for Integrating Third-Party Network Devices
 - Overview of Integrating Network Devices into the SRC Network
 - Logging In Subscribers and Creating Sessions

Setting Up SAE Communities

Tasks to configure SAE communities are:

- If there is a firewall in the network, configuring the firewall to allow SAE messages through.
1. Adding Virtual Router Objects on page 87
 2. Configuring the SAE Community Manager on page 88
 3. Specifying the Community Manager in the SAE Device Driver on page 89

Adding Virtual Router Objects

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 configuration 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 ...]
```

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-plug-in [ tracking-plug-in ...]
```

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;
```

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
```

2. 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
```

3. 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
```

4. 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
```

5. 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
```

6. (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:

1. 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;
```

Configuring SAE Properties for the Event Notification API with 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 ■ Initially Configuring the SAE

Developing Router Initialization Scripts for JUNOSe Routers, JUNOS Routing Platforms, and Network Devices

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 `IorPublisher` script in the `/opt/UMC/sae/lib` folder. The `IorPublisher` 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 4 on page 91 describes the router initialization scripts that we provide with the SRC software in the `/opt/UMC/sae/lib` folder.

Table 6: 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 JUNOSe 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 JUNOSe 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 5 on page 92 describes the fields that the SAE exports.

Table 7: 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 SsperrorLog.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
```

Copying Initialization Scripts to the C-series Controller

If you use a script that is not provided with the SRC software, 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:
```

Specifying Initialization Scripts on the SAE

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
```

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 ■ Setting Up SAE Communities

Using the NIC Resolver in Environments that have Third-Party Devices (C-Web Interface)

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 Configuring the NIC (SRC CLI)

Part 4

Locating Subscriber Management Information

- Locating Subscriber Information with the NIC on page 97
- Configuring NIC (SRC CLI) on page 113
- Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrflp on page 135
- Configuring Applications to Communicate with an SAE on page 147
- Configuring SRC Applications to Communicate with an SAE (SRC CLI) on page 149
- Developing Applications That Use NIC on page 157
- NIC Resolution Process on page 165
- NIC Configuration Scenarios on page 171

Chapter 8

Locating Subscriber Information with the NIC

- Locating Subscriber Management Information on page 97
- Mapping Subscribers to a Managing SAE on page 99
- High Availability for NIC on page 100
- Planning a NIC Implementation on page 103
- NIC Configuration Scenarios on page 103
- NIC Agents Used in the NIC Configuration Scenarios on page 109
- Router Initialization Scripts with NIC Configuration Scenarios on page 111

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.

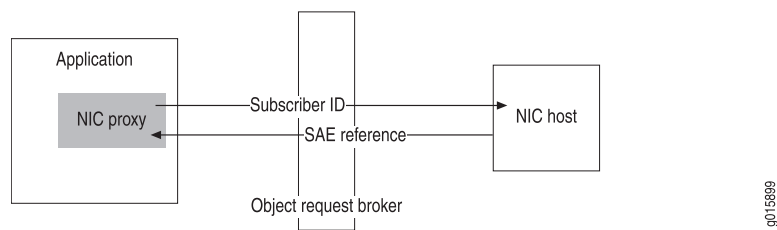
NIC can operate in a client/server mode or in a local host mode. In the client/server mode, a NIC host and NIC proxies can reside on different systems. In local host mode, a NIC host and NIC proxies reside in the same process on a machine.

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 98 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

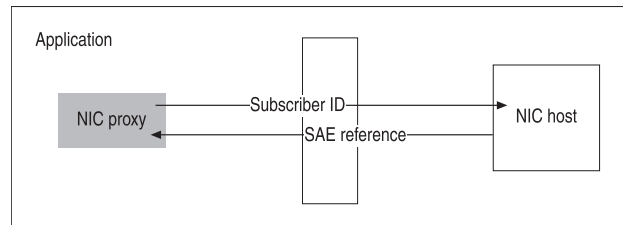
In local host mode, a Java application can include the libraries for a NIC host as well as NIC proxies. With this configuration, the NIC host and the NIC proxies communicate with each other within the same application. Because both components run within the same application, the application and the NIC 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 99 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

9015957

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.

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 8 on page 100 describes the types of agents supplied with NIC.

Table 8: 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.
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

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.

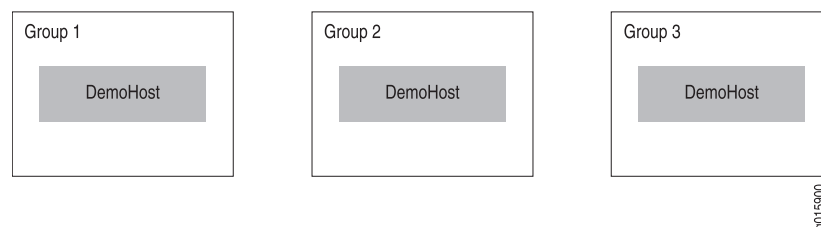
NIC Replication

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 101 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

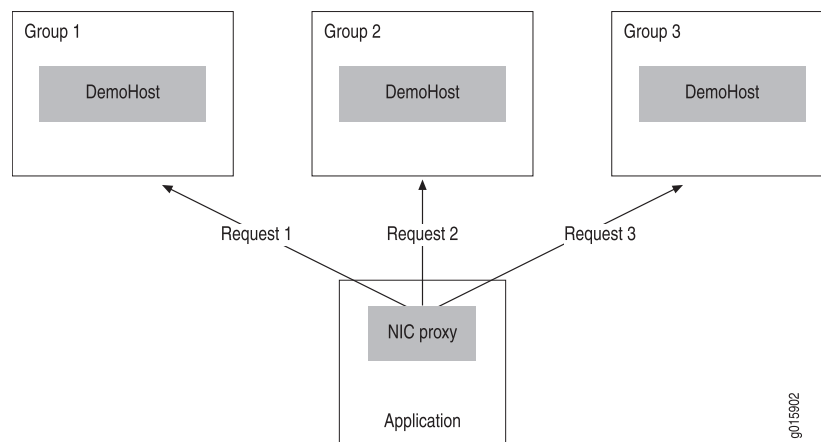
With NIC replication, a NIC proxy can contact multiple NIC hosts that are assigned to different groups. When a NIC proxy is configured to contact more than one group,

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 102 shows resolution requests sent by means of a round-robin selection.

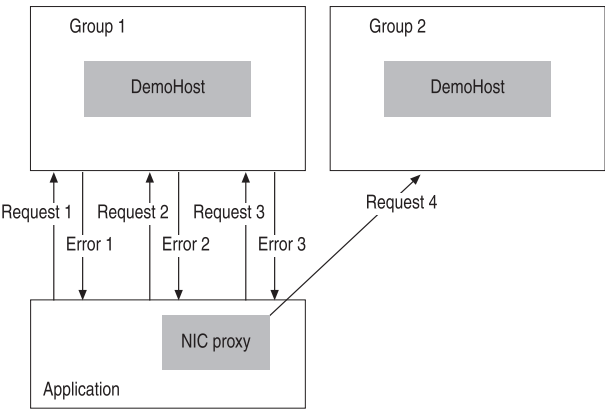
Figure 13: NIC Group Selection by Round-Robin



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 103 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



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.

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.
- 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.

NIC Configuration Scenarios

Table 9 on page 104 lists the NIC configuration scenarios provided in the SRC software.

Table 9: 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.	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.
Sample use:			
DSL providers for residential customers.			
For subscribers who have an accounting ID. Can be used for multiple subscribers who use the same accounting ID, in which case NIC returns all SAE IORs for mapped subscribers.	OnePopAcctId	Accounting ID of a subscriber to the SAE IOR and the IP address of a subscriber to accounting ID	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. For example, customers can assign a subscriber username (login id without domain name) to an accounting ID with the following subscriber classification. [< -retailerDn- > ?accountingUserId = < -userName- > ?sub?(uniqueID = < -userName- >)]
Sample use:			
Support for the volume-tracking application.			

Table 9: NIC Configuration Scenarios *(continued)*

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
For subscribers who have assigned IP addresses (assigned external to the SAE).	OnePopDynamicIp	Subscriber IP address to the SAE IOR	
Sample use: In a PacketCable Multimedia Specification (PCMM) environment when the SAE acts as both a policy server and application manager.			
For resolution of a subscriber login name to an SAE IOR, and of a subscriber IP address to a subscriber login name.	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.
Sample use: Support for tracking subscriber bandwidth usage or for using a billing model. You can use the SRC-VTA with this scenario.			
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 9: 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

Table 9: NIC Configuration Scenarios *(continued)*

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
<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 9: NIC Configuration Scenarios *(continued)*

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
<p>For scenarios in which subscribers have an assigned IP address and these IP addresses can be associated with interfaces on JUNOS routing platforms.</p> <p>Sample use:</p> <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.
<p>For scenarios in which subscribers have an assigned IP address.</p> <p>Sample use:</p> <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	<p>Similar to OnePopStaticRouteIp. Used to support multiple VPNs with overlapping IP pools.</p> <p>Static route information for routers resides in an XML document in the directory under the router object.</p>
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			

Table 9: NIC Configuration Scenarios *(continued)*

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
For two POPs that share a back office.	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.
Sample use:			
Support for a deployment that has a back office that connects to NIC hosts at other sites.			You can add POPs by copying the configuration for one POP and modifying the configuration to suit your environment.

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 10 on page 109 lists all agents that are available in the various configuration scenarios.

Table 10: 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.

Table 10: NIC Agents *(continued)*

Agent Name	Type of Agent	Type of Information
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.
VrSaeId	Directory	Reads information about virtual routers and the mappings between virtual routers and SAEs.

Table 11 on page 110 shows the types of agents that each configuration scenario uses.

Table 11: Agents in Configuration Scenarios

NIC Configuration Scenario	Directory Agents	SAE Plug-In Agents	SAE Client Agents
OnePop	PoolVr, VrSaeId		
OnePopAcctId	PoolVr, VrSaeId	AcctIdIp, IpAcctId	
OnePopDnSharedIp	PoolVr, VrSaeId, Enterprise	DnVr	
OnePopDynamicIp	PoolVr, VrSaeId		
OnePopLogin	Pool, VrSaeId	IpLoginName, LoginNameVr	
OnePopLoginPull			IpSaeId, LoginSaeId
OnePopPcmm	PoolVr, VrSaeId		
OnePopSharedIp	PoolVr, VrSaeId	IpVr	
MultiPop	PoolVr, VrSaeId, site-specific versions of PoolVr and VrSaeId	IpVr	
OnePopAllRealms	PoolVr, VrSaeId, Enterprise	IpVr	
OnePopPrimaryUser	VrSaeId	UserNameVr	
OnePopStaticRouteIp	VrSaeId, PoolInterface		
OnePopVrflp	VrSaeId, PoolInterface		

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 12 on page 111 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 12: 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 with the CLI.

For more information about router initialization scripts for JUNOS routing platforms, see [Configuring the SAE to Manage JUNOS Routing Platforms](#).

Chapter 9

Configuring NIC (SRC CLI)

- Configuration Statements for the NIC on page 113
- Before You Configure the NIC on page 115
- Configuring the NIC (SRC CLI) on page 116
- Starting the NIC (SRC CLI) on page 117
- Reviewing and Changing Operating Properties for NIC (SRC CLI) on page 117
- Configuring NIC Replication (SRC CLI) on page 119
- Configuring a NIC Scenario (SRC CLI) on page 120
- Configuring Advanced NIC Features on page 130
- Verifying Configuration for the NIC (SRC CLI) on page 130
- Testing a NIC Resolution (SRC CLI) on page 130
- Stopping a NIC Host on a C-series Controller (SRC CLI) on page 131
- Restarting the NIC (SRC CLI) on page 131
- Restarting a NIC Agent (SRC CLI) on page 132
- Restarting a NIC Resolver (SRC CLI) on page 132
- Changing NIC Configurations (SRC CLI) on page 133

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: 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-plugin {
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** ■ 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](#).

- Ensure that the appropriate type of router initialization script is configured for the router or network device.

See [Locating Subscriber Management Information](#).

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**
- [Locating Subscriber Management Information](#)
 - [Configuring the NIC \(SRC CLI\)](#)

Configuring the NIC (SRC CLI)

Before you configure the NIC, complete the prerequisite tasks.

See [Before You Configure the NIC](#).

To configure the NIC:

1. Start the NIC component.

See [Starting the NIC \(SRC CLI\)](#).

2. Configure NIC operating properties.

See [Reviewing and Changing Operating Properties for NIC \(SRC CLI\)](#).

3. Configure NIC replication.

See [Reviewing and Changing Operating Properties for NIC \(SRC CLI\)](#).

4. (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 \(SRC CLI\)](#).

5. Configure a NIC scenario.

See [Configuring a NIC Scenario \(SRC CLI\)](#).

6. Verify the NIC configuration.

See [Verifying Configuration for the NIC \(SRC CLI\)](#).

- Related Topics**
- Changing NIC Configurations (SRC CLI)
 - Testing a NIC Resolution (SRC CLI)

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**
- Configuring the NIC (SRC CLI)

Reviewing and Changing Operating Properties for 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 117
2. Changing NIC Operating Properties on page 118

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.

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.
- 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 with the SRC CLI

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 120
- Configuring Directory Agents on page 123
- Configuring SAE Client Agents on page 125
- Configuring SAE Plug-In Agents on page 127
- Configuring the SAE to Communicate with SAE Plug-In Agents When You Use NIC Replication on page 128

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 \(SRC CLI\)](#).

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
...
```

2. From configuration mode, access the statement that configures a NIC configuration scenario, and specify the name of a scenario.

```
[edit]
user@host# edit shared nic scenario name
```

For example:

```
[edit]
user@host# edit shared nic scenario OnePopLogin
```

3. View the default configuration for the configuration scenario. For example:

```
[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-plug-in {

```

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 VrSaeld 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 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 ;
}
```

```
}
```

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 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
```

- 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-plug-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:

- 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
```

- 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.

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
```

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.

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**
- Configuring the NIC (SRC CLI)
 - Testing a NIC Resolution (SRC CLI)

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:
0000000000000354944C3A738D67742E6A756E697065722E6E65742F7361652F5365727669636541637469766174696F6E456E67696E653A312E3000
00000000000100000000000068000102000000000F3137322E32382E3233302E313230000022610000000001073726320382F736165504F412F53
41450000002000000000000008000000004441430000000010000001C0000000000100010000000105010001001090000000105010001
user@host>
```

The following example shows an unsuccessful resolution for an IP key that has the value 192.168.8.2:

```
user@host> test nic resolve locator /nicLocators/ip key 192.168.3.2
Failed to resolve key 192.168.3.2 for resolver /nicLocators/ip due to
net.juniper.smgmt.gateway.nic.protocol.NICExc
IDL:net/juniper/smgmt/gateway/nic/protocol/NICExc:1.0
user@host>
```

Related Topics ■ Configuring NIC Test Data (SRC CLI)

Stopping a NIC Host on a C-series Controller (SRC CLI)

If you run NIC in client/server mode, you can stop the NIC host independently of the NIC proxy.

To stop a NIC host:

- From operational mode, disable the NIC.

```
user@host> disable component nic
```

Related Topics ■ Restarting the NIC (SRC CLI)
 ■ Restarting a NIC Agent (SRC CLI)
 ■ Restarting a NIC Resolver (SRC CLI)

Restarting the NIC (SRC CLI)

To restart a NIC host:

- From operational mode, restart the NIC.

```
user@host> request restart nic
```

You can also restart the NIC at the slot level.

- Related Topics**
- Stopping a NIC Host on a C-series Controller (SRC CLI)
 - Restarting a NIC Agent (SRC CLI)
 - Restarting a NIC Resolver (SRC CLI)

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)
 - Restarting the NIC (SRC CLI)
 - Restarting a NIC Resolver (SRC CLI)

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)
 - Restarting the NIC (SRC CLI)
 - Restarting a NIC Agent (SRC CLI)

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)

Chapter 10

Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrflp

- Overview of the Network Publisher on page 135
- NIC Document That Maps Subscriber IP Addresses to a JUNOS Interface on page 136
- Configuration Statements for the Network Publisher on page 136
- Before You Configure and Run the Network Publisher on page 137
- Configuring the Network Publisher on page 138
- Running the Network Publisher on page 143
- Overview of Files to Test Network Publisher on page 144
- Configuring Information to Test the Network Publisher on page 144
- Troubleshooting Network Publisher Operations on page 145
- Reviewing the Information Collected from a JUNOS Routing Platform on page 146

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.
- OnePopVrflp—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
 - Overview of Files to Test Network Publisher
 - Before You Configure and Run the Network Publisher
 - Configuration Statements for the Network Publisher

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
 - Reviewing the Information Collected from a JUNOS Routing Platform

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
 - Configuring the Network Publisher

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 OnePopVrfIp configuration scenario.
- Before you run the network publisher, make sure that the NIC is enabled.

Related Topics

- Overview of the Network Publisher
- Configuration Statements for the Network Publisher
- Configuring the Network Publisher
- Starting the NIC (SRC CLI)

Configuring the Network Publisher

To configure the network publisher, complete the following tasks:

1. Configuring Local Configuration for the Network Publisher on page 138
2. Configuring Connections Between JUNOS Routing Platforms and the Network Publisher on page 139
3. Configuring Router Authentication for the Network Publisher on page 140
4. Configuring Routing Table Filters for the Network Publisher on page 141
5. Configuring the Connection Between the Network Publisher and the Juniper Networks Database on page 142

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
```

Running the Network Publisher

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

- Overview of the Network Publisher
- Before You Configure and Run the Network Publisher
- Configuring the Network Publisher
- Starting the NIC (SRC CLI)

Overview of Files 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
 - Configuring Information to Test the Network Publisher
 - Reviewing the Information Collected from a JUNOS Routing Platform

Configuring Information to Test the Network Publisher

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**
- Overview of the Network Publisher
 - Overview of Files to Test Network Publisher

■ Troubleshooting Network Publisher Operations

Troubleshooting Network Publisher Operations

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](#)

5. Configure the network publisher to use an output file, and review the file.

See [Reviewing the Information Collected from a JUNOS Routing Platform](#)

- Related Topics**
- [Overview of the Network Publisher](#)
 - [Before You Configure and Run the Network Publisher](#)
 - [Configuring the Network Publisher](#)

Reviewing the Information Collected from a JUNOS Routing Platform

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](#)
 - [Overview of Files to Test Network Publisher](#)
 - [Troubleshooting Network Publisher Operations](#)
 - [Specifying Filenames and URLs](#)

Chapter 11

Configuring Applications to Communicate with an SAE

- Overview of NIC Proxy Configuration on page 147
- Before You Configure a NIC Proxy on page 147

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*.

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*, *Overview of the NIC Resolution Process*, and *Overview of NIC Proxy Configuration*.



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.

Chapter 12

Configuring SRC Applications to Communicate with an SAE (SRC CLI)

- Configuration Statements for NIC Proxies on page 149
- Configuring Resolution Information for a NIC Proxy (SRC CLI) on page 150
- Changing the Configuration for the NIC Proxy Cache (SRC CLI) on page 151
- Configuring a NIC Proxy for NIC Replication (SRC CLI) on page 152
- Configuring NIC Test Data (SRC CLI) on page 154

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** ■ For detailed information about each configuration statement, see *SRC-PE CLI Command Reference*.

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
```

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
```

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
```

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
```


Chapter 13

Developing Applications That Use NIC

- External Application Requirements for NIC on page 157
- External Non-Java Applications That Use NIC on page 157
- Creating a NIC Locator to Include with a Non-Java Application on page 158
- External Java Applications That Use NIC on page 158
- Developing a Java Application to Communicate with a NIC Proxy on page 159
- Updating Information About Address Pools on page 164

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.

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>.

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

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>.

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 160
- Passing a Reference to the Configuration Manager to the NIC Factory on page 160
- Instantiating the NIC Factory Class on page 160
- Initializing Logging on page 161
- Instantiating the NIC Proxy on page 161
- Managing a Resolution Request on page 162
- Deleting Invalid Results from the NIC Proxy's Cache on page 163
- Removing the NIC Proxies on page 164

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*.

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](#).

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](#).

- `constValue`—Specific value of the constraint.

For the allowed syntax for the data types, see [Overview of the NIC Resolution Process](#).

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.

- 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
 - 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.

Chapter 14

NIC Resolution Process

- Overview of the NIC Resolution Process on page 165
- NIC Data Types on page 166
- Constraints as NIC Data Types on page 169

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 13 on page 165 shows the resolutions that the components in the NIC configuration scenarios perform. For customized types of resolutions, contact Juniper Networks Professional Services.

Table 13: 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 and Constraints as NIC Data Types](#).

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 > # < vrfId > @vrName@routerName
- Example—FastEthernet4/1.0/4@boston@router1
 fastEthernet4/1.0#vpn_a@boston@router1

Interfaceld

- 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])

Saeld

- 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.

Chapter 15

NIC Configuration Scenarios

- Overview of NIC Configuration Scenarios on page 171
- OnePop Scenario on page 172
- OnePopPcmm Scenario on page 174
- OnePopDynamicIp Scenario on page 176
- OnePopSharedIp Scenario on page 178
- OnePopStaticRouteIp on page 180
- OnePopVrflp Scenario on page 182
- OnePopAcctId Scenario on page 185
- OnePopLogin Scenario on page 186
- OnePopLoginPull Scenario on page 189
- OnePopPrimaryUser on page 189
- OnePopDnSharedIp Scenario on page 191
- OnePopAllRealms Scenario on page 195
- MultiPop Scenario on page 199

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](#).

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 172 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.

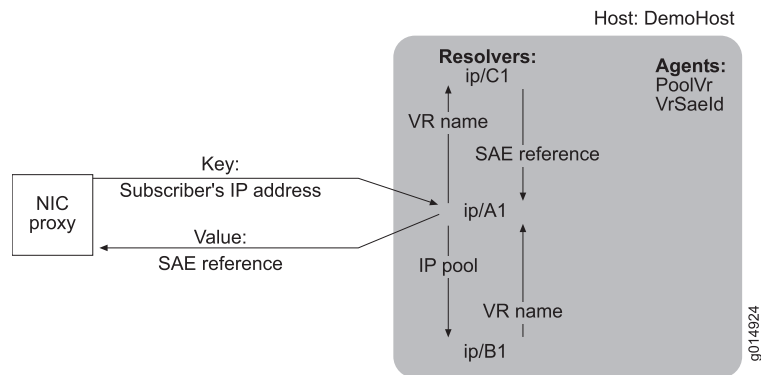
The OnePop sample provides two host configurations: a centralized configuration and a distributed configuration. The OnePop Centralized configuration also provides an example of NIC host redundancy.

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.

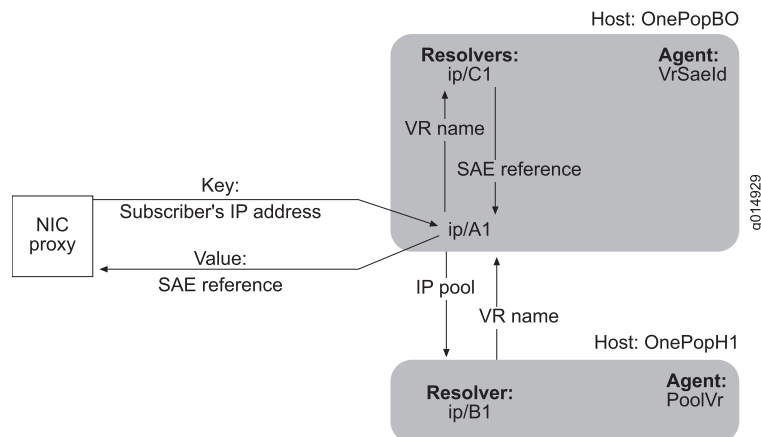
Figure 16 on page 173 shows the interactions of the NIC components for this realm.

Figure 16: OnePop Centralized Configuration

Distributed Configuration

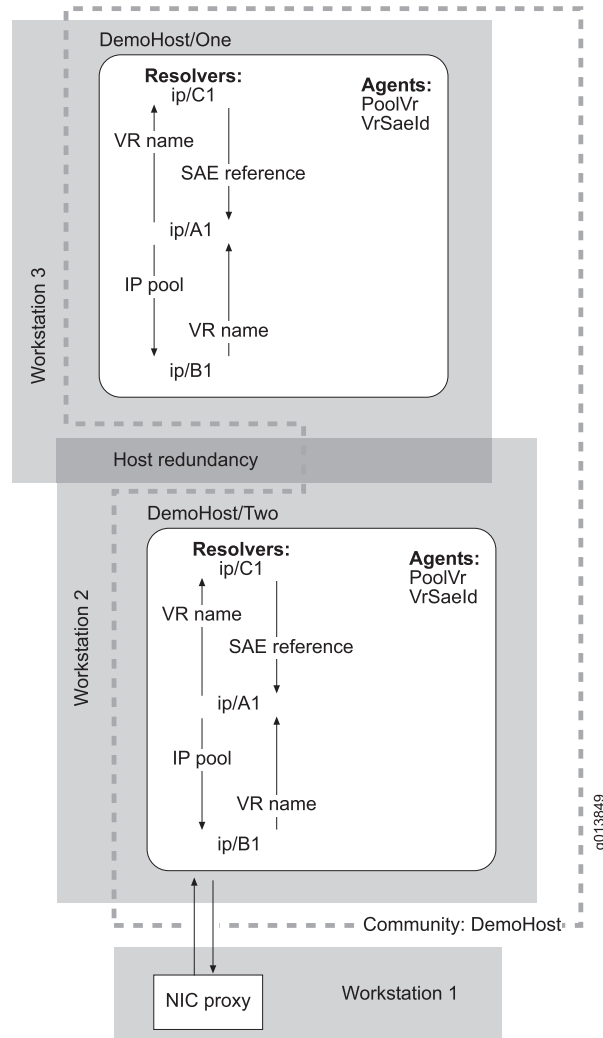
In this configuration, the agents and resolvers are distributed among several hosts. When the NIC proxy sends a subscriber's IP address to host OnePopBO, the components execute the same actions as they do in the centralized configuration.

Figure 17 on page 173 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

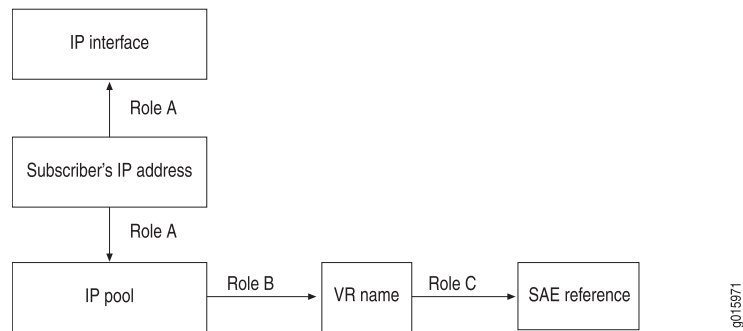
OnePopPcmm Scenario

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 175 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 > .

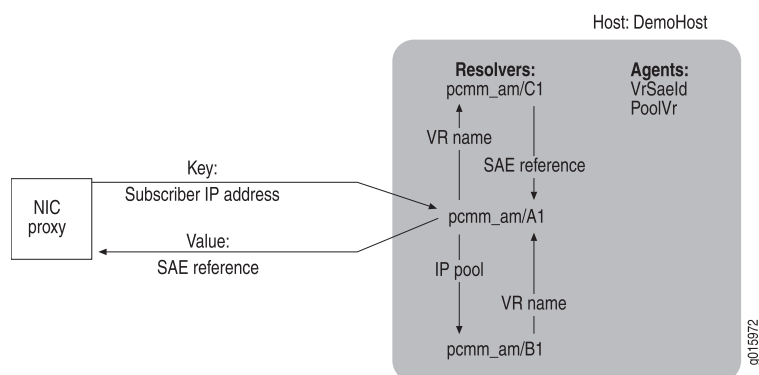
The OnePopPcomm scenario provides two host configurations: a centralized configuration and a distributed configuration.

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.

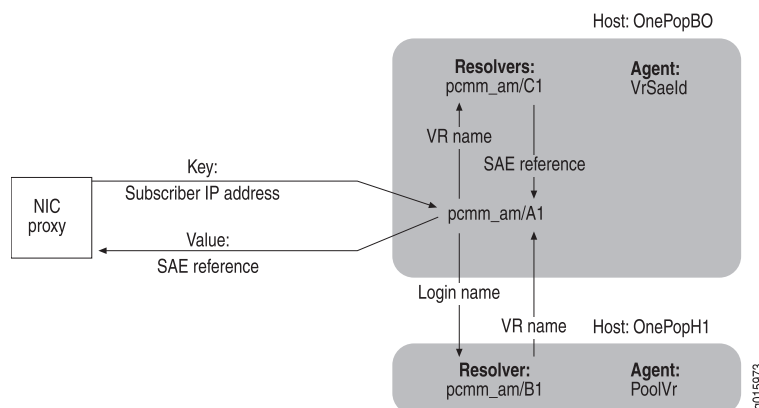
Figure 20 on page 176 show the interactions of the NIC components for this realm.

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 176 illustrates the interactions of the NIC components for this realm.

Figure 21: OnePopPcmm Distributed Configuration

OnePopDynamicIp Scenario

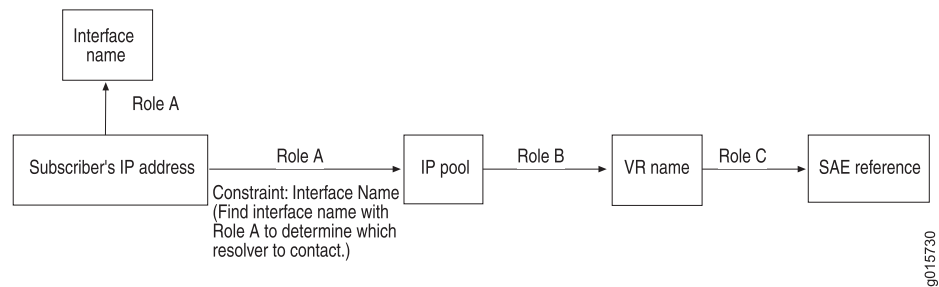
This scenario illustrates a configuration that is very similar to the OnePop scenario. The realm for this configuration accommodates the situation in which IP address pools are configured locally on each virtual router object. The resolution process takes a subscriber's IP address as the key and returns 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 177 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.

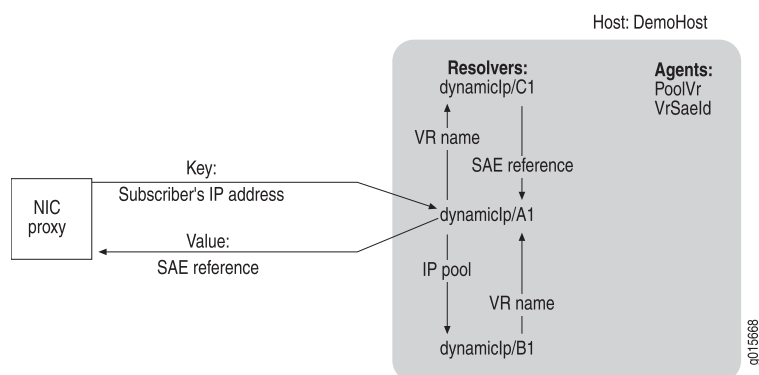
The OnePopDynamicIp scenario provides two host configurations: a centralized configuration and a distributed configuration.

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.

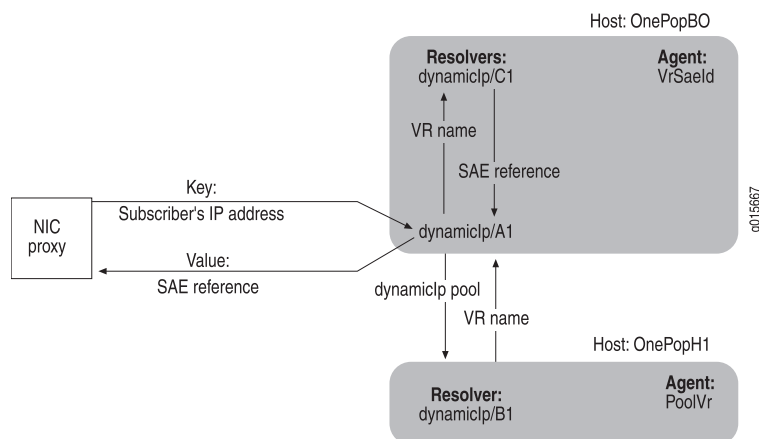
Figure 23 on page 178 illustrates the interactions of the NIC components for this realm.

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.

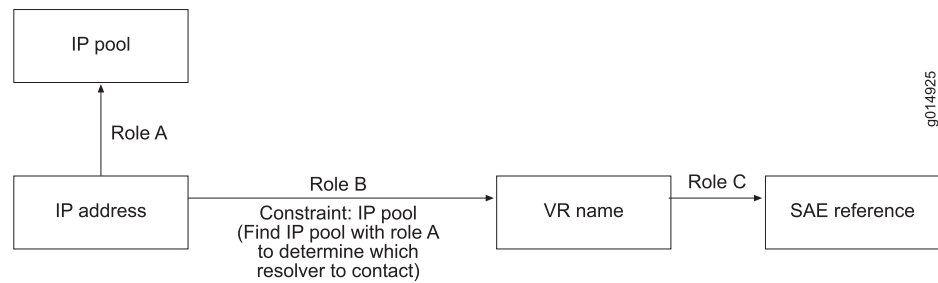
Figure 24 on page 178 illustrates the interactions of the NIC components for this realm.

Figure 24: OnePopDynamicIp Distributed Configuration

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 179 shows the resolution graph for this realm.

Figure 25: Resolution Process for sharedIp Realm

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.

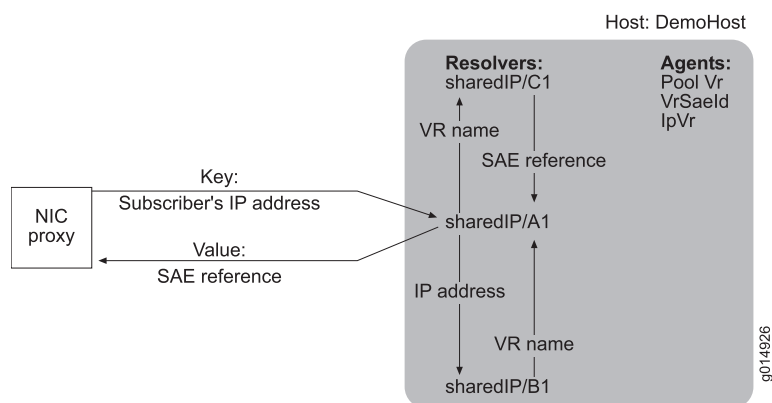
The OnePopSharedIP scenario provides two host configurations: a centralized configuration and a distributed configuration.

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:

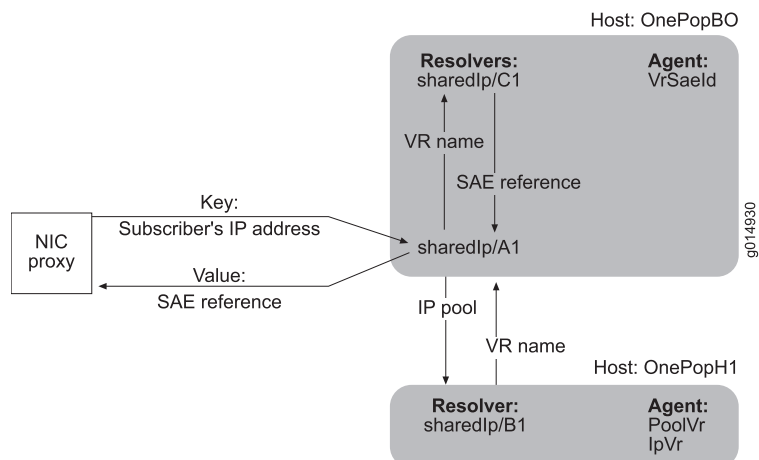
1. The host passes the IP address to resolver A1.
2. Resolver A1 obtains an IP pool for the IP address.
3. Resolver A1 forwards the IP address and the IP pool to resolver B1.
4. Resolver B1 obtains a VR name for the IP address and returns the VR name to resolver A1.
5. Resolver A1 forwards the VR name 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 180 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 180 illustrates the interactions of the NIC components for this realm.

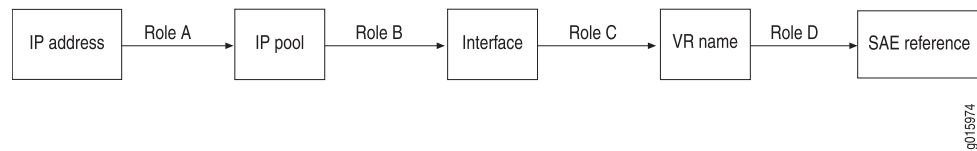
Figure 27: OnePopSharedIP Distributed Configuration

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 181 shows the resolution graph for this realm.

Figure 28: Resolution Process for the StaticRouteIp 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.

The OnePopStaticRouteIp scenario provides two host configurations: a centralized configuration and a distributed configuration.

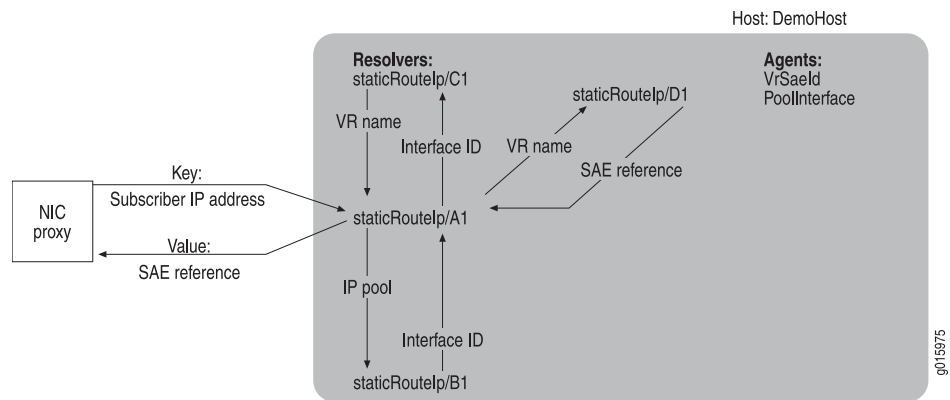
Centralized Configuration

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 182 shows the interactions of the NIC components for this realm.

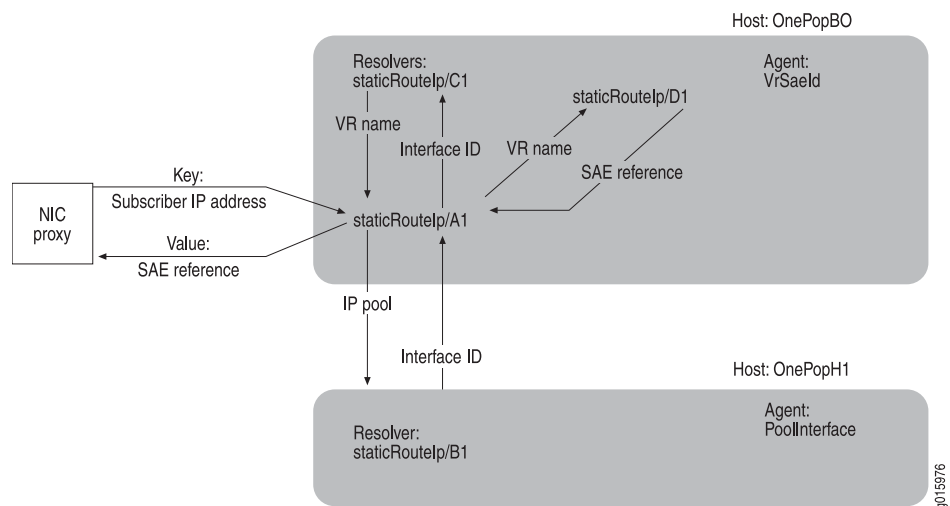
Figure 29: OnePopStaticRouteIp Centralized Configuration



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 182 illustrates the interactions of the NIC components for this realm.

Figure 30: OnePopStaticRouteIp Distributed Configuration



OnePopVrflp Scenario

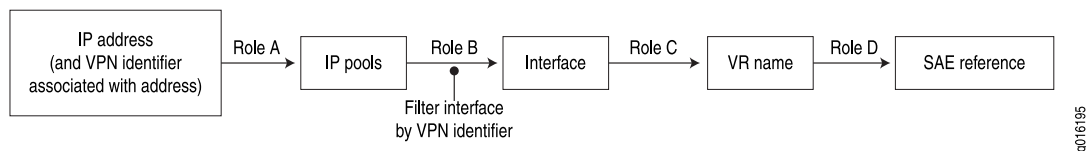
The OnePopVrflp configuration scenario for NIC resolves an assigned IP address for a subscriber to IP pools or network whose traffic enters the network through an interface on a JUNOS routing platform to a reference for the SAE that manages the interface. The realm for this configuration utilizes routing 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 RouteIp scenario. During resolution, the OnePopVrflp scenario filters interfaces the VPN identifier of the VPN that carries subscriber traffic.

Figure 31 on page 183 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 Overview of Files to Test Network Publisher.

The OnePopVrflp scenario provides two host configurations: a centralized configuration and a distributed configuration.

Centralized Configuration

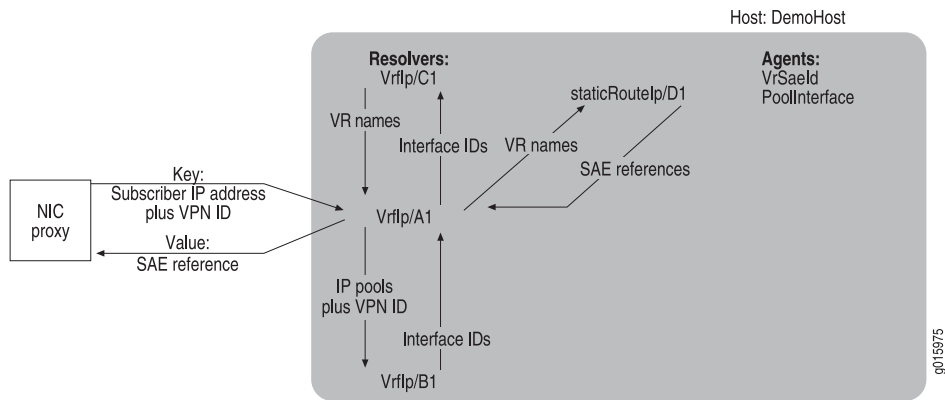
In this configuration, the single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of events occurs:

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 184 shows the interactions of the NIC components for this realm.

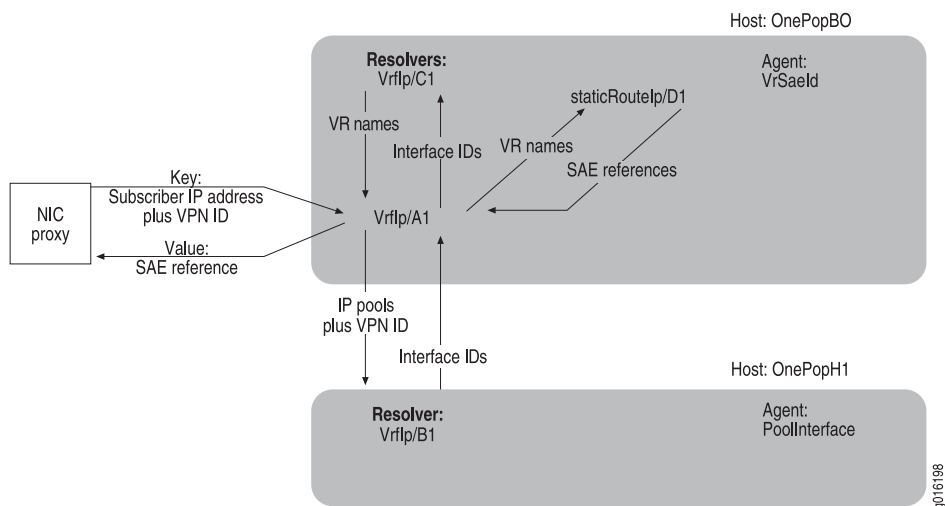
Figure 32: OnePopVrflp Centralized Configuration



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 184 illustrates the interactions of the NIC components for this realm.

Figure 33: OnePopStaticRoutelp Distributed Configuration

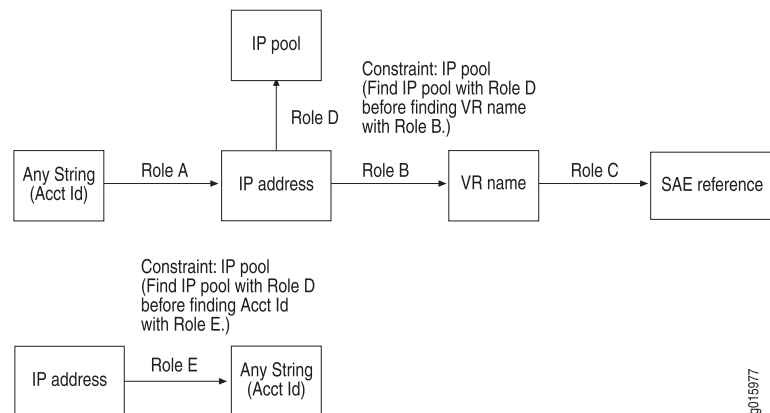


OnePopAcctId Scenario

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 185 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.

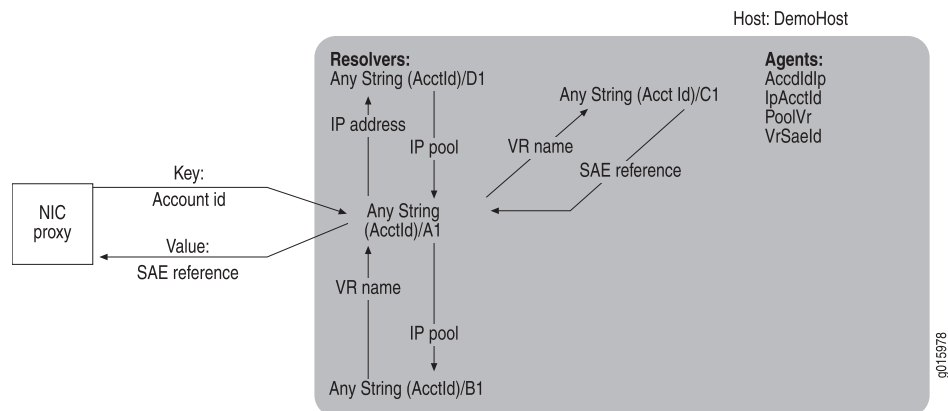
5. Resolver A1 forwards the IP address and IP pool to Resolver B1.
6. Resolver B1 obtains the VR name and return it to resolve A1.
7. Resolver A1 forwards the VR name to resolver C1.
8. Resolver C1 obtains the SAE reference for the VR name and returns it to resolver A1.
9. Resolver A1 passes the SAE 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 186 illustrates the interactions of the NIC components for this realm.

Figure 35: OnePopAcctId Centralized Configuration

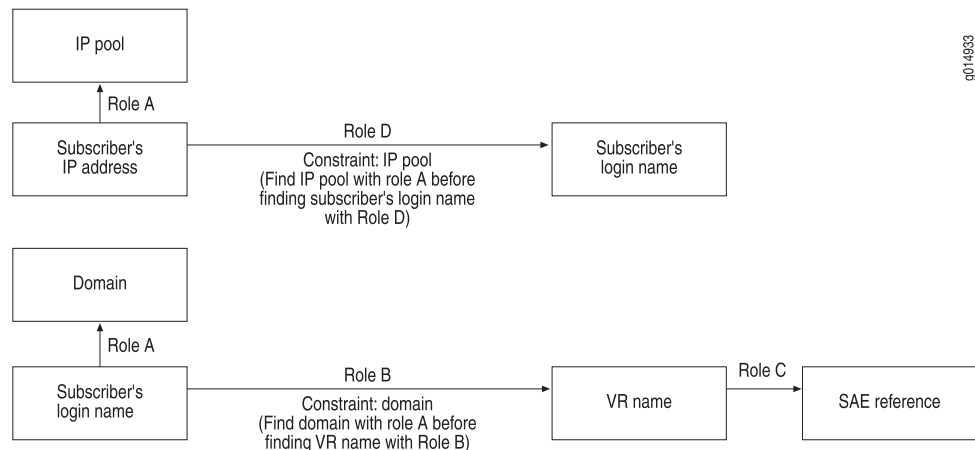


OnePopLogin Scenario

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 187 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.

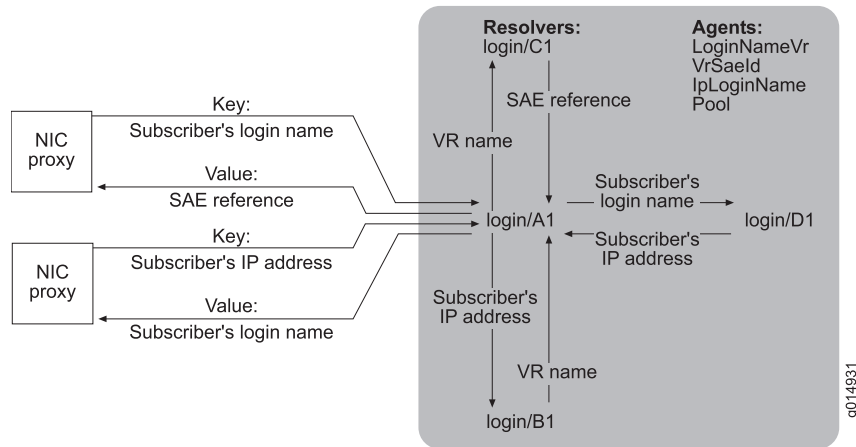
4. Resolver B1 obtains a VR name for the login name and returns the VR name to resolver A1.
5. Resolver A1 forwards the VR name to resolver C1.
6. Resolver C1 obtains an SAE reference for the VR and returns the SAE reference to resolver A1.
7. Resolver A1 returns the SAE reference to its host.
8. The 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.

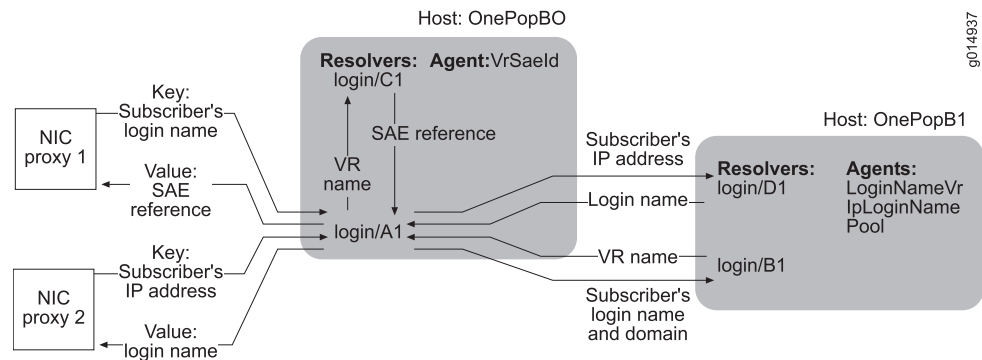
Figure 37 on page 188 illustrates the interactions of the NIC components for this realm.

Figure 37: OnePopLogin Centralized Configuration



Distributed Configuration

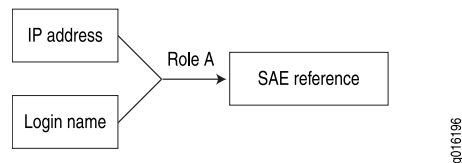
In this configuration, the agents and resolvers are distributed among several hosts. When the NIC proxy sends a subscriber's IP address to the host OnePopBO, the resolvers execute the same actions as they do in the centralized configuration. Figure 38 on page 189 illustrates the interactions of the NIC components for this realm.

Figure 38: OnePopLogin Distributed Configuration

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 189 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.

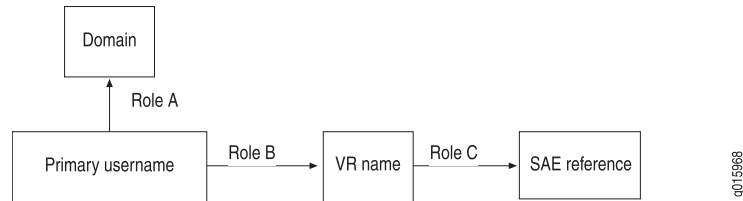
The OnePopLoginPull scenario includes the following SAE client agents:

- LoginSaeId
- IpSaeId

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 190 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.

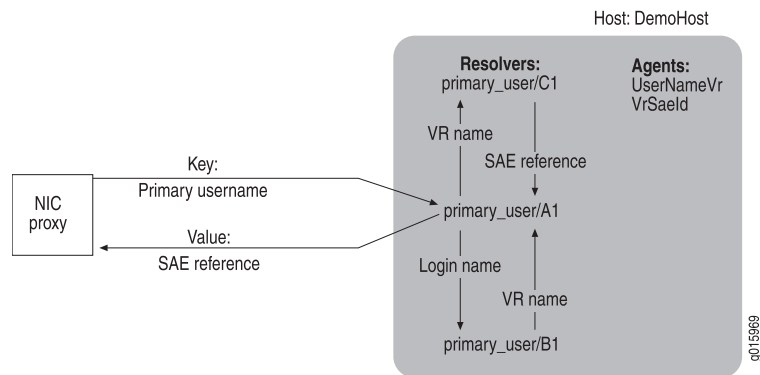
The OnePopPrimaryUser scenario provides two host configurations: a centralized configuration and a distributed configuration.

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:

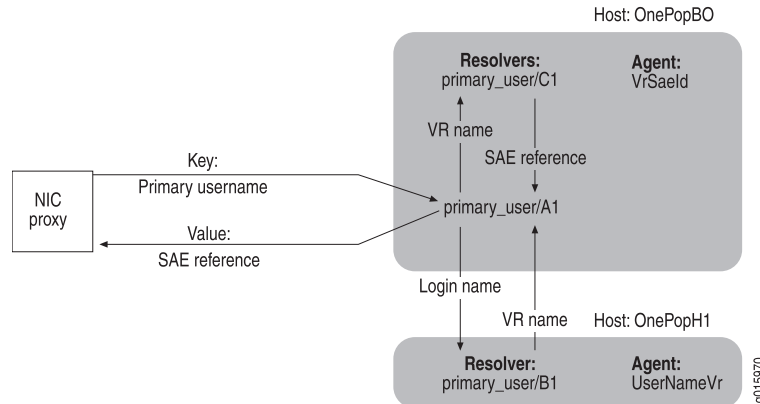
1. The host passes the primary username to resolver A1.
2. (Optional) Resolver A1 resolves the primary username to its domain.
3. Resolver A1 forwards the primary username to resolver B1.
4. Resolver B1 obtains the name of the VR associated with the subscriber's primary username and returns the VR to 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 191 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 191 illustrates the interactions of the NIC components for this realm.

Figure 42: OnePopPrimaryUser Distributed Configuration

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 OnePopShared scenario (see Figure 25 on page 179).

- DN realm

This realm uses essentially the same resolution process as the MultiPop DN realm (see Figure 50 on page 203). 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 193 illustrates the centralized configuration, and Figure 44 on page 195 illustrates the distributed configuration for the DN realms.

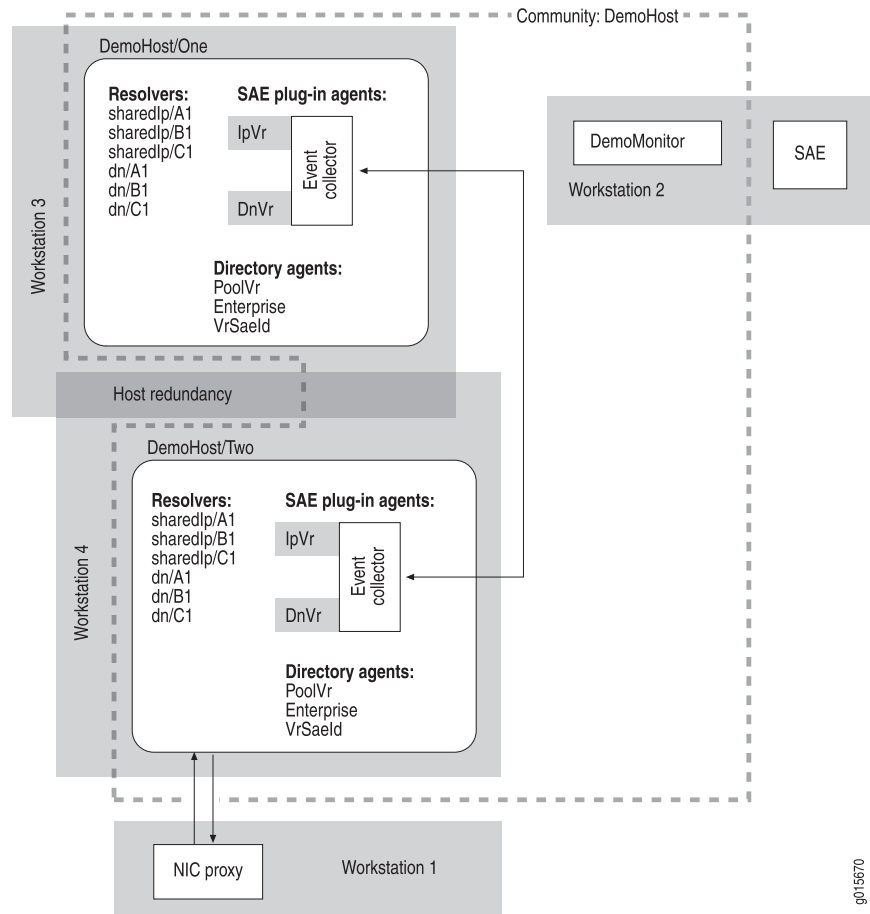
The configuration for the two realms is similar to the configuration for the shared IP and DN realms in the OnePopAllRealms scenario. .

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 193 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 195 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.

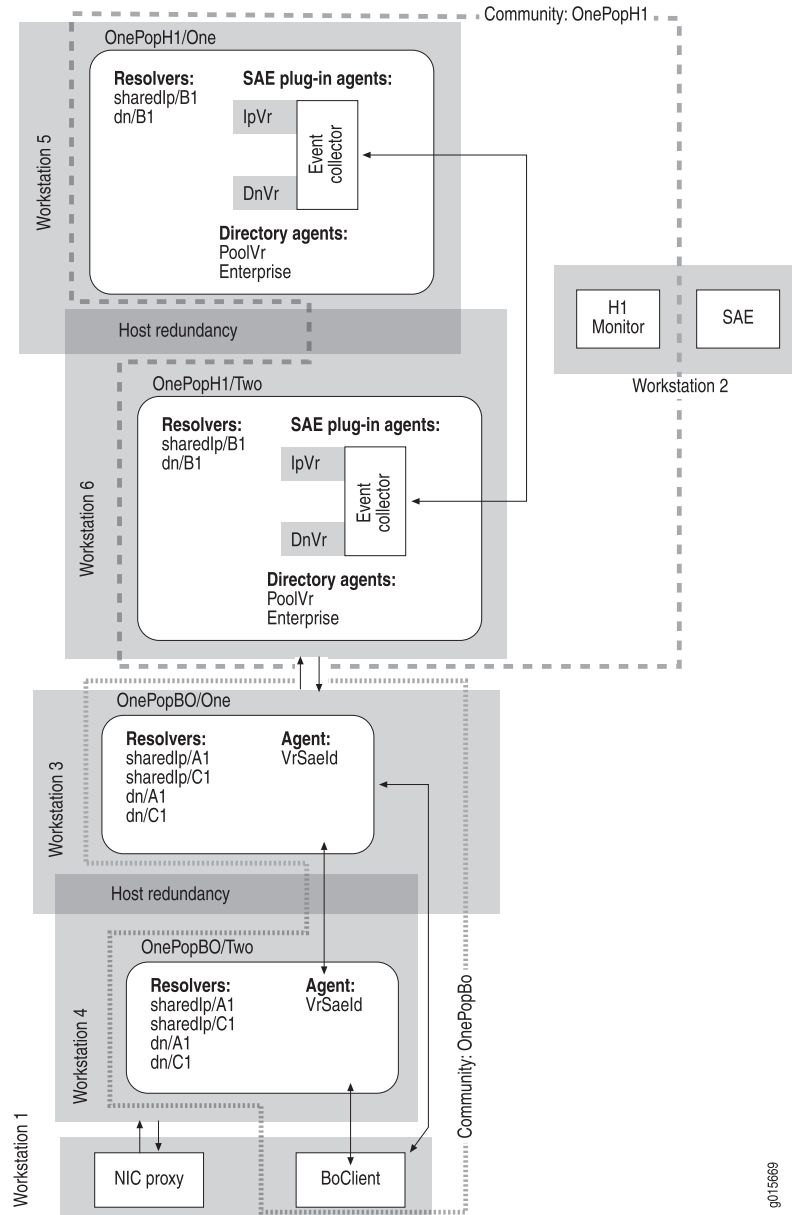
The redundant hosts for OnePopH1 (OnePopH1/One and OnePopH1/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host.

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, H1 Monitor, 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.

H1 Monitor 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

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 (see Figure 15 on page 172). However, some of the constraints differ.

- Shared IP

The resolution process is identical to that for the OnePopShared scenario (see Figure 25 on page 179).

- DN realm

This realm uses essentially the same resolution process as the MultiPop DN realm (see Figure 50 on page 203). However, some of the constraints differ.

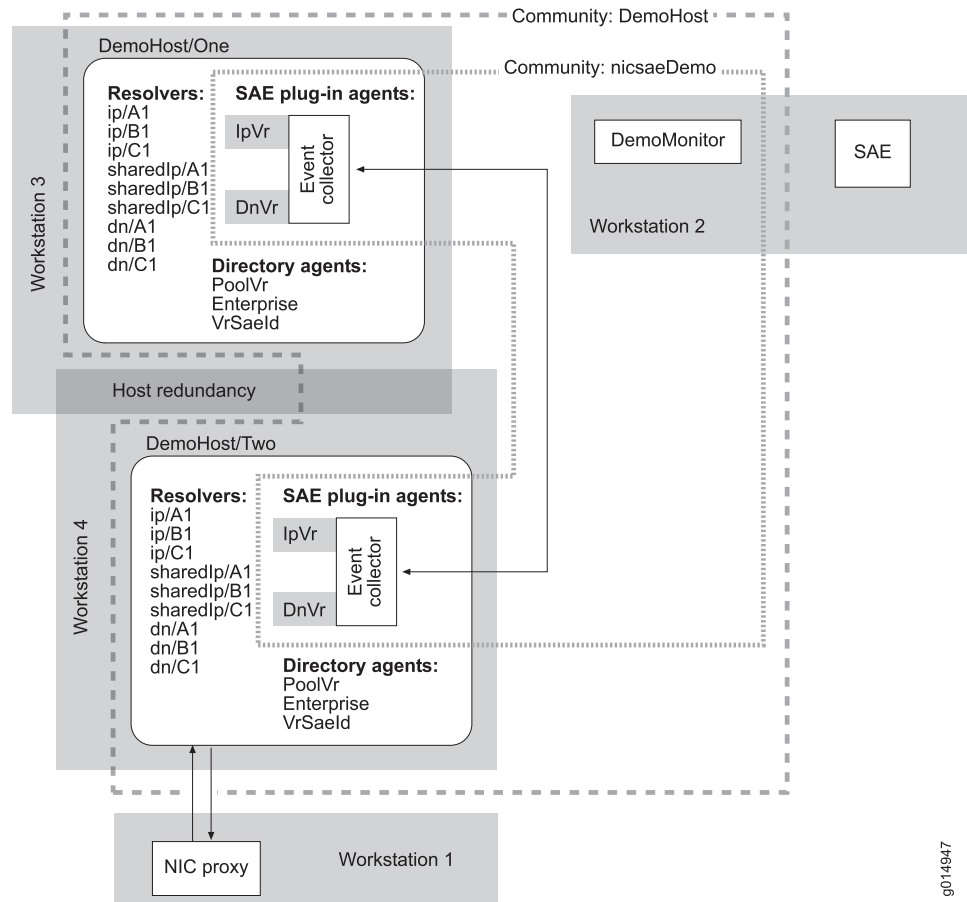
This realm also uses the same agents as the MultiPop DN realm. The names of agents and resolvers are essentially the same as those in the MultiPop configuration, although they do not include a POP identifier. By reviewing the scenario, Figure 45 on page 197 and Figure 46 on page 199, you can determine exact pictures of the DN realms for the centralized and distributed configurations.

Figure 45 on page 197 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 199 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.

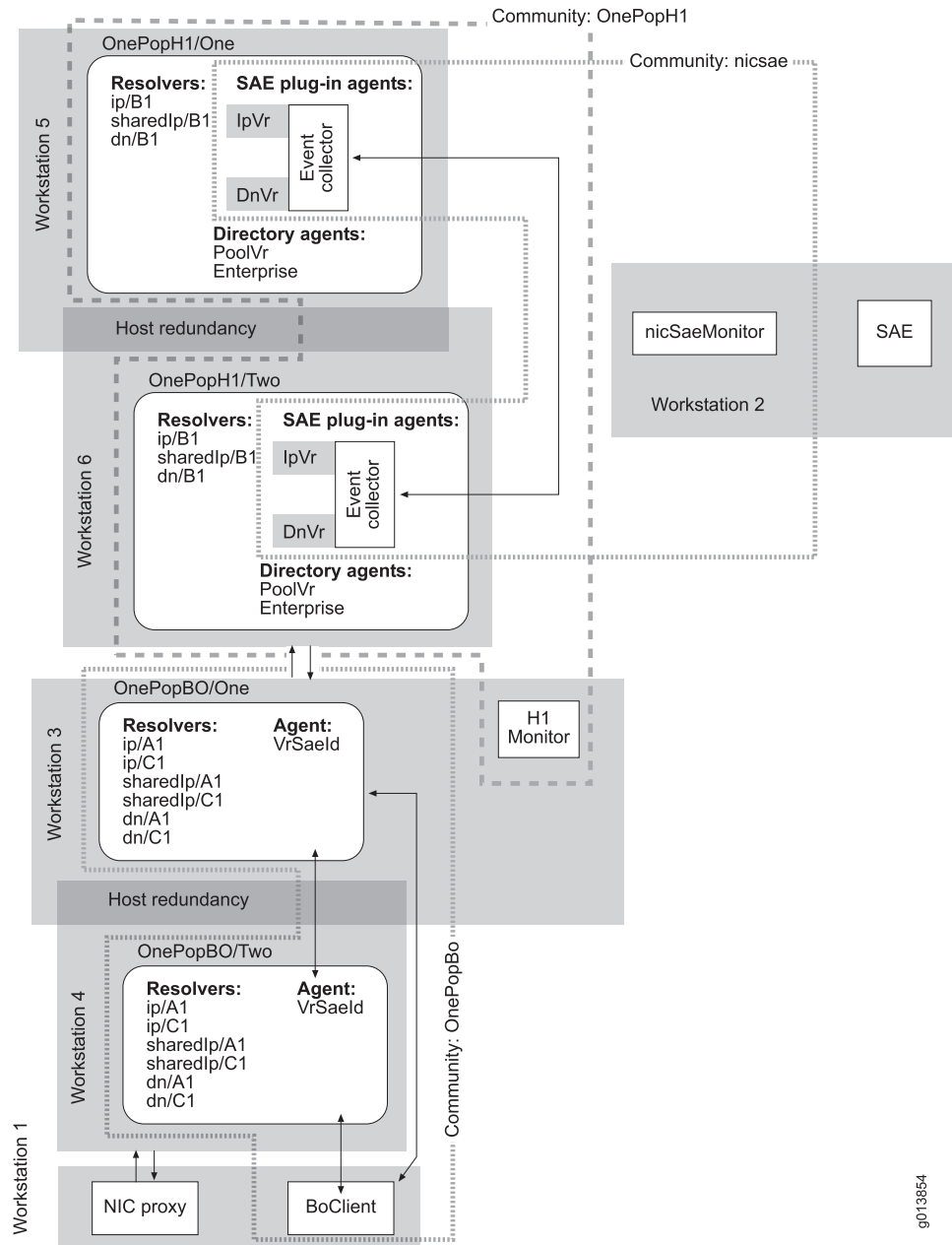
The redundant agents form a community called nicsae with monitor nicSaeMonitor, which tracks them. The redundant agents are identified in the community by the

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, H1 Monitor, 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.

H1 Monitor 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

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.

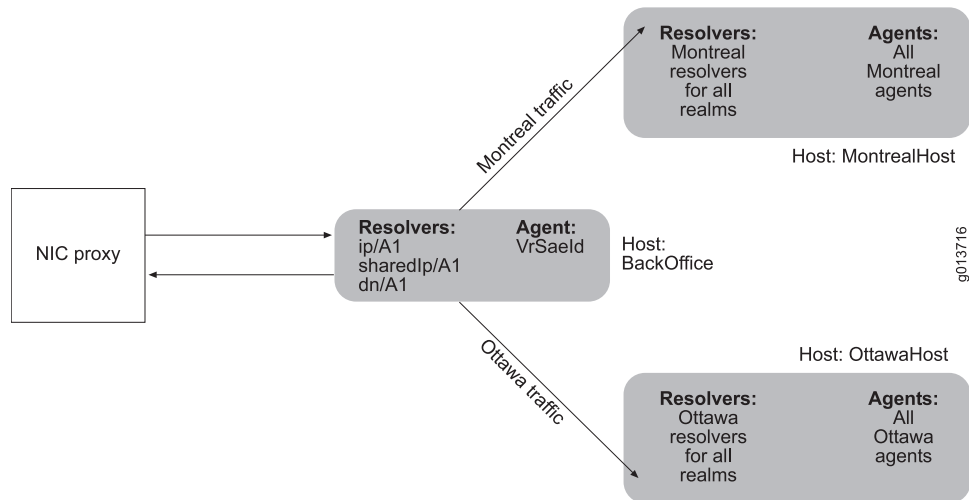
The scenario shows three realms for this configuration:

- IP
- Shared IP
- DN

Each realm provides a different type of resolution. The following sections provide information about these realms.

Figure 47 on page 200 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 Figure 15 on page 172). However, some of the constraints differ.

The following agents interact with the resolvers in this realm:

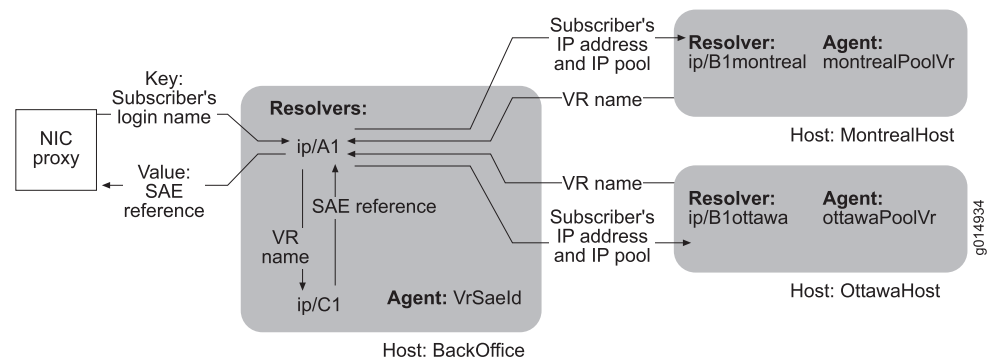
- Directory agents `montrealPoolVr` and `ottawaPoolVr` collect and publish information that maps IP address pools to VRs. Each agent publishes only the information that is relevant to its POP. You achieve selective publishing by relating an Ottawa scope to the VRs in the Ottawa POP and a Montreal scope 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 201 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. Figure 16 on page 173 shows the resolution graph for this realm.

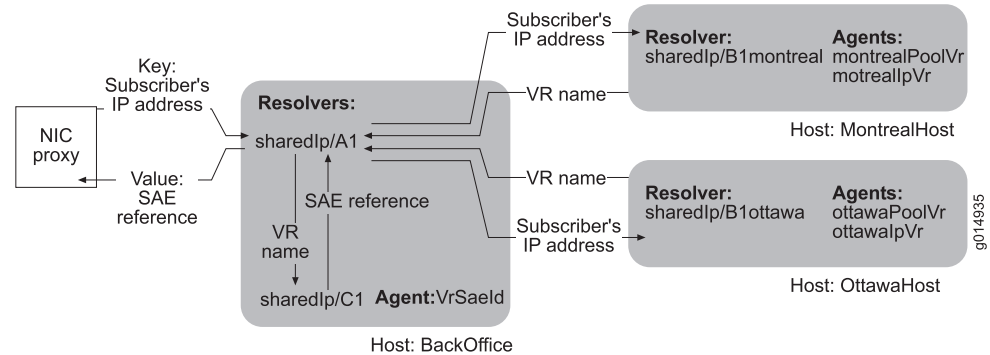
The following agents interact with resolvers in this realm:

- 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.

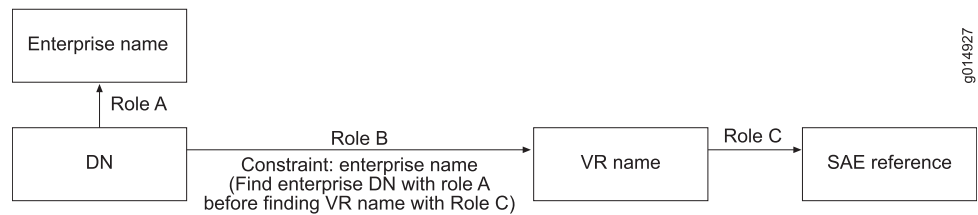
Figure 49 on page 203 illustrates the interactions of the NIC components for this realm.

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 203 shows the resolution process for this realm.

Figure 50 on page 203 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.

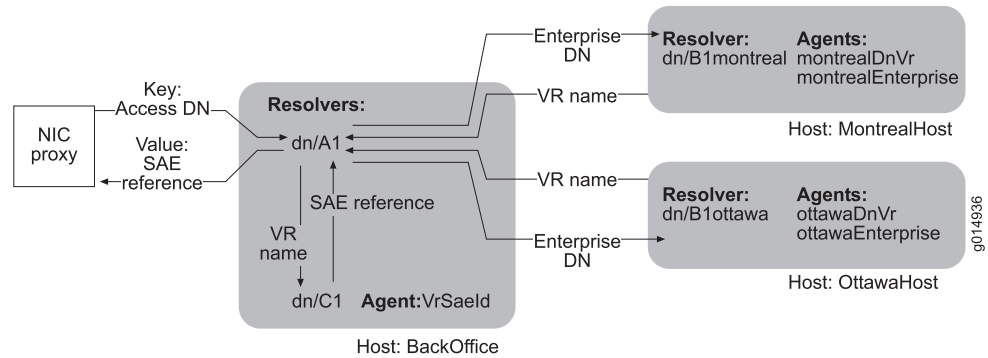
When the NIC proxy sends an access DN to host BackOffice, the following sequence of events occurs:

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 204 illustrates the interactions of the NIC components for this realm.

Figure 51: dn Realm for MultiPop Configuration



Part 5

Providing Admission Control with SRC-ACP

- Overview of Providing Admission Control with SRC-ACP on page 207
- Configuring Admission Control (SRC CLI) on page 217
- Configuring Congestion Point Classification (SRC CLI) on page 253
- Managing SRC-ACP (SRC CLI) on page 263
- Monitoring Admission Control (SRC CLI) on page 265
- Monitoring Admission Control (C-Web Interface) on page 275

Chapter 16

Overview of Providing Admission Control with SRC-ACP

- Overview of SRC-ACP on page 207
- Deriving Congestion Points Automatically on page 209
- Allocating Bandwidth to Applications Not Controlled by SRC-ACP on page 211
- Use of Multiple SRC-ACPs on page 212
- Interactions Between SRC-ACP and Other Components on page 212
- Redundancy on page 213
- Fault Recovery on page 214
- State Synchronization on page 214
- API for ACP on page 215

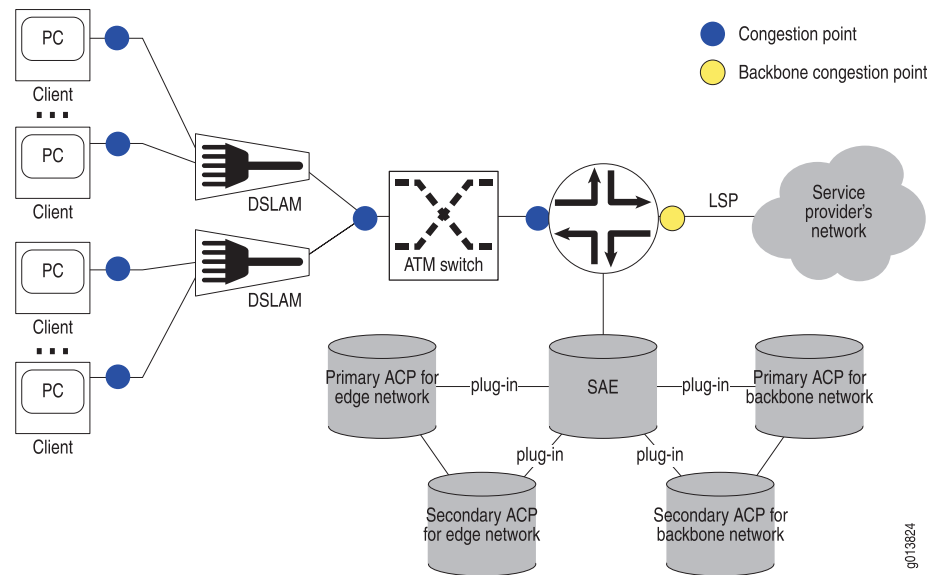
Overview of SRC-ACP

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 208 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 API for ACP. 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.

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 14 on page 209 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 14: Congestion Points Derived Through NAS Port ID

Congestion Points	Location of Object in Directory
Physical interface on router	<code>interfaceName = ATM < slot > / < port > , orderedCimKeys = < routerName > , o = AdmissionControl, o = umc</code> < slot > —Number of port on router < port > —Number of port on router < routerName > —Hostname configured for router

Table 14: Congestion Points Derived Through NAS Port ID *(continued)*

Congestion Points	Location of Object in Directory
ATM virtual path	interfaceName = ATM < slot > / < port > : < vpi > <i>orderedCimKeys = < routerName > , o = AdmissionControl, o = umc</i> < vpi > —Number of virtual path on router
ATM virtual connection	interfaceName = ATM < slot > / < port > : < vpi > . < vci > <i>orderedCimKeys = < routerName > , o = AdmissionControl, o = umc</i> < 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**
- For more information about automatically deriving congestion points from a configured congestion point profile, see *Deriving Congestion Points Automatically*.
 - *Defining a Congestion Point Profile*
 - *Configuring SRC-ACP*

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.

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.

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 .)

Redundancy

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 208). 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.

The primary and secondary SRC-ACPs communicate with each other through a CORBA interface. When you start each SRC-ACP (see Starting SRC-ACP), it will register its redundancy CORBA interface with the naming service application, and import the interface for the other SRC-ACP from the naming service application.

Each SRC-ACP continuously monitors the other's availability. If the primary SRC-ACP becomes unavailable, the secondary SRC-ACP immediately notifies the naming service application and assumes the primary role. If the former primary SRC-ACP recovers very quickly, it will again assume the primary role. However, if the former primary SRC-ACP recovers more slowly, it will assume the primary role only if the former secondary SRC-ACP becomes unavailable.

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.

State Synchronization

You can configure SRC-ACP to synchronize states with the SAE.

If state synchronization is enabled, the current state can 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.

Both SRC-ACP redundancy and state synchronization can be enabled at the same time. 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.

API for 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 attr)

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.iior` 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.

Chapter 17

Configuring Admission Control (SRC CLI)

- Configuration Statements for SRC-ACP on page 217
- Configuring SRC-ACP on page 219
- Creating Grouped Configurations for SRC-ACP on page 220
- Configuring Local Properties for SRC-ACP on page 221
- Configuring the SAE for SRC-ACP on page 224
- Configuring SRC-ACP Properties on page 227
- Configuring SRC-ACP to Manage the Edge Network on page 239
- Configuring SRC-ACP to Manage the Backbone Network on page 242

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

To use SRC-ACP in the SRC network, you must perform some configuration. For information about these configuration procedures, see:

1. (Optional) Creating Grouped Configurations for SRC-ACP
2. Configuring Local Properties for SRC-ACP
3. Configuring the SAE for SRC-ACP
4. Configuring SRC-ACP Properties
5. (Edge and dual mode only) Configuring SRC-ACP to Manage the Edge Network
6. (Backbone and dual mode only) Configuring SRC-ACP to Manage the Backbone Network
7. Starting SRC-ACP

You can automate and scale the configuration of congestion points using congestion point classification. For more information, see [Classifying Congestion Points](#).

Creating Grouped Configurations for SRC-ACP

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](#).

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](#).

Configuring Local Properties for SRC-ACP

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 221](#)
- [Configuring Initial Properties for SRC-ACP on page 222](#)
- [Configuring Directory Connection Properties for SRC-ACP on page 223](#)
- [Configuring Initial Directory Eventing Properties for SRC-ACP on page 223](#)

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](#).

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;

```

Configuring the SAE for SRC-ACP

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 224](#)
- [Configuring Event Publishers on page 225](#)
- [Configuring the SAE to Monitor Interfaces for Congestion Points on page 225](#)

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*.

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*).

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).

```
[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
```

Configuring SRC-ACP Properties

To configure SRC-ACP properties, perform these tasks:

1. Configuring Logging Destinations for SRC-ACP on page 227
2. Configuring SRC-ACP Operation on page 228
3. Configuring CORBA Interfaces on page 232
4. Configuring SRC-ACP Redundancy on page 233
5. Configuring Connections to the Subscribers' Directory on page 234
6. Configuring Connections to the Services' Directory on page 236
7. Configuring SRC-ACP Scripts and Classification on page 237

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 with 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 with 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.
- < 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
```

Configuring SRC-ACP to Manage the Edge Network

The tasks to configure SRC-ACP to manage the edge network are:

- Configuring Network Interfaces in the Directory for the Edge Network on page 239
- Configuring Bandwidths for Subscribers on page 240
- Assigning Network Interfaces to Subscribers on page 241
- Configuring Bandwidths for Services in the Edge Network on page 242

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.

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).

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).

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
```

Configuring SRC-ACP to Manage the Backbone Network

The tasks to configure SRC-ACP to manage the backbone network are:

- Configuring Network Interfaces in the Directory for the Backbone Network on page 243
- Extending SRC-ACP Congestion Points for the Backbone Network on page 243

- Configuring Action Congestion Points on page 244
- Configuring Bandwidths for Services in the Backbone Network on page 245
- Configuring Congestion Points for Services in the Backbone Network on page 245
- Using Functions for Backbone Congestion Point Classification Scripts on page 249
- Configuring Congestion Point Profiles in the Directory on page 250
- Assigning Interfaces to Congestion Point Profiles on page 251

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*.

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.

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. For information about the attributes that can be embedded in the expression, see Plug-In Attributes for Use with Backbone Congestion Point Expressions.

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.

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 JUNOS routers: interfaceName = “ fastEthernet6/0”
 For JUNOS routing platforms: interfaceName = “fe-0/1/0.0”
 For forwarding interface: interfaceName = “FORWARDING_INTERFACE”

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

- Port identifier of an interface.
- Value—Includes interface name and additional layer 2 information
- Example—nasPort = “ fastEthernet 3/1 ” (There is a space between fastEthernet and slot number 3/1 in the nasPort field.)

portId

- Identifier of VLAN or virtual circuit.
- Value—String; for a virtual circuit, use the format <VPI> / <VCI>

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
```


Chapter 18

Configuring Congestion Point Classification (SRC CLI)

- Overview of Congestion Point Classification on page 253
- Configuration Statements for Congestion Point Classification on page 254
- Classifying Congestion Points on page 254
- Defining a Congestion Point Profile on page 259
- Congestion Point Expressions on page 260

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.

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*.

Classifying Congestion Points

The tasks to classify congestion points are:

1. Configuring Targets and Criteria for Classification Scripts on page 254
2. Configuring Classification Scripts Contents for Classification Scripts on page 255
3. Configuring Congestion Point Classification Targets on page 255

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.

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.

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 JUNOSe 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 JUNOSe 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"`

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

- Port identifier of an interface.
- Value—Includes interface name and additional layer 2 information
- Example—nasPort = “ fastEthernet 3/1 ” (There is a space between fastEthernet and slot number 3/1 in the nasPort field.)

portId

- Identifier of VLAN or virtual circuit.
- Value—String; for a virtual circuit, use the format <VPI> / <VCI>

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.

Defining a Congestion Point Profile

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 .

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 260 and “Match Criteria for Congestion Point Classification” on page 261.

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, V.

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.

Chapter 19

Managing SRC-ACP (SRC CLI)

Topics in this chapter include:

- Starting SRC-ACP on page 263
- Stopping SRC-ACP on page 263
- Reorganizing the File That Contains ACP Data on page 263
- Modifying Congestion Points on page 263

Starting SRC-ACP

To start SRC-ACP:

```
user@host> enable component acp
```

Stopping SRC-ACP

To stop SRC-ACP:

```
user@host> disable component acp
```

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
```

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
```

Chapter 20

Monitoring Admission Control (SRC CLI)

- Viewing Information About Subscriber Sessions in the Edge Network on page 265
- Viewing Edge Congestion Point Information by DN on page 266
- Viewing Edge Congestion Point Information by Subscriber Session on page 267
- Viewing Information About Services in the Backbone Network on page 267
- Viewing Backbone Congestion Point Information by DN on page 268
- Viewing Backbone Congestion Point Information by Service on page 268
- Viewing Action Congestion Point Information by Service on page 269
- Viewing Action Congestion Point Information by Congestion Point on page 270
- Viewing Information About Subscribers Obtained from External Applications on page 271
- Viewing Congestion Point Information by DN on page 272
- Viewing Congestion Point Information by Name on page 272
- Viewing SNMP Information for Devices on page 273
- Viewing SNMP Information for the Directory on page 273
- Viewing SNMP Information for SRC-ACP on page 273

Viewing Information About Subscriber Sessions in the Edge Network

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.

Viewing Edge Congestion Point Information by DN

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
```

Viewing Edge Congestion Point Information by Subscriber Session

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
```

Viewing Information About Services in the Backbone Network

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.

Viewing Backbone Congestion Point Information by DN

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.

Viewing Backbone Congestion Point Information by Service

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.

Viewing Action Congestion Point Information by Service

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.

Viewing Action Congestion Point Information by Congestion Point

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 DNSs:

```
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.

Viewing Information About Subscribers Obtained from External Applications

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.

Viewing Congestion Point Information by DN

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.

Viewing Congestion Point Information by Name

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.

Viewing SNMP Information for Devices

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.

Viewing SNMP Information for the Directory

Purpose View SNMP information for the directory.

Action To display statistics for directory SNMP information:

```
user@host> show acp statistics directory
```

Viewing SNMP Information for SRC-ACP

Purpose View SNMP information for SRC-ACP.

Action To display statistics for SRC-ACP SNMP information:

```
user@host> show acp statistics general
```


Chapter 21

Monitoring Admission Control (C-Web Interface)

- Viewing Information About Subscriber Sessions in the Edge Network (C-Web Interface) on page 275
- Viewing Information About Edge Congestion Points by DN on page 276
- Viewing Information About Edge Congestion Points by Subscriber Session on page 277
- Viewing Information About Services in a Backbone Network (C-Web Interface) on page 278
- Viewing Information About Congestion Points in a Backbone Network by Expression on page 280
- Viewing Information About Congestion Points in a Backbone Network by DN on page 281
- Viewing Information about Action Congestion Points in a Backbone Network by Service on page 282
- Viewing Information about Action Congestion Points in a Backbone Network by Expression on page 283
- Viewing Information About Subscribers Obtained from External Applications (C-Web Interface) on page 285
- Viewing Information About Congestion Points from an External Application by DN on page 286
- Viewing Information About Congestion Points from an External Application by Interface Name on page 286
- Viewing Statistics for the SRC-ACP Configuration (C-Web Interface) on page 287

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.

Viewing Information About Edge Congestion Points by DN

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.

ACP		Edge / 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 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.

Viewing Information About Edge Congestion Points by Subscriber Session

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.	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: Display congestion point attributes	detail
Virtual Router Name	Name of virtual router from which to list congestion points.	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 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.

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								
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 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, 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 .

Viewing Information About Congestion Points in a Backbone Network by Expression

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.

The screenshot shows the 'Congestion Point Expression' configuration pane. The top navigation bar includes 'Monitor', 'Configure', 'Diagnose', and 'Manage'. The user is logged in as 'admin'. The left sidebar lists various configuration categories: ACP, CLI, Component, Date, Disk, Interfaces..., Iptables..., JPS, NIC, NTP, Redirect Server, Route..., SAE, Security, and System. The main pane is titled 'ACP' and 'Backbone / Congestion Point / Congestion Point Expression'. It contains several input fields with corresponding descriptions and default values:

Field	Description	Value	Default
Interface Alias	Interface alias used by backbone service to generate congestion points.	Interface alias	No value
Interface Description	Description of interface used by backbone service to generate congestion points.	Interface description	No value
Interface Name	Name of interface related to congestion points.	Interface name	No value
Nas Port Id	Interface NAS port ID used by backbone service to generate congestion points.	NAS port ID	No value
Service Name	Name of service used by backbone service to generate congestion points.	Service name	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: Display congestion point attributes	detail
Virtual Router Name	Name of virtual router from which to list congestion points.	Virtual router name	No value

At the bottom of the pane are 'OK' and 'Reset' buttons.

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](#).

Viewing Information About Congestion Points in a Backbone Network by DN

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.

Monitor	Configure	Diagnose	Manage
ACP			
CLI			
Component			
Date			
Disk			
Interfaces...			
Iptables...			
JPS			
NIC			
NTP			
Redirect Server			
Route...			
SAE			
Security			
System			

Logged in as: admin Refresh Preferences About Logout

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

OK 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.

Viewing Information about Action Congestion Points in a Backbone Network by Service

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.

Field	Description
Interface Alias	Interface alias used by backbone service to generate congestion points. <i>Value:</i> Interface alias <i>Default:</i> No value
Interface Description	Description of interface used by backbone service to generate congestion points. <i>Value:</i> Interface description <i>Default:</i> No value
Interface Name	Name of interface related to congestion points. <i>Value:</i> Interface name <i>Default:</i> No value
Nas Port Id	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	Name of service used by backbone service to generate congestion points. <i>Value:</i> Service name <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 backbone service attributes <i>Default value:</i> detail
Virtual Router Name	Name of virtual router from which to list backbone services. <i>Value:</i> Virtual router name <i>Default:</i> 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 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.

Viewing Information about Action Congestion Points in a Backbone Network by Expression

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.

The screenshot shows the 'ACP' configuration pane for 'Backbone / Congestion Point / Congestion Point Expression'. The interface includes a top navigation bar with 'Monitor', 'Configure', 'Diagnose', and 'Manage' tabs. Below this is a sidebar with a tree view containing 'ACP', 'CLI', 'Component', 'Date', 'Disk', 'Interfaces...', 'Iptables...', 'JPS', 'NIC', 'NTP', 'Redirect Server', 'Route...', 'SAE', 'Security', and 'System'. The main area displays the configuration form for 'ACP'.

ACP		
Backbone / Congestion Point / Congestion Point Expression		
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 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.

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.	Choices: brief, detail	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.

Viewing Information About Congestion Points from an External Application by DN

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.

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.

Viewing Information About Congestion Points from an External Application by Interface Name

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.

The screenshot shows the Juniper C-Web Interface. The top navigation bar includes 'Monitor', 'Configure', 'Diagnose', and 'Manage'. The user is logged in as 'admin'. The left sidebar shows a tree view with 'ACP' selected. The main content area is titled 'Remote Update / Congestion Point / Name'. It contains four configuration rows: 'Device Name' (text input), 'Interface Name' (text input), 'Slot' (text input), and 'Style' (dropdown menu). Each row has a description and default value. At the bottom are 'OK' and 'Reset' buttons. The footer shows copyright information for Juniper Networks, Inc. 2007.

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 Interface Name box, enter the interface name of the congestion point, or leave the box blank to display information about all interfaces.
4. In the Slot box, enter the number of the slot for which you want to display congestion point information.
5. Select an output style from the Style list.
6. Click **OK**.

The Remote Update/Congestion Point/Name pane displays the congestion points.

Viewing Statistics for the SRC-ACP Configuration (C-Web Interface)

- Viewing General Statistics for SRC-ACP on page 287
- Viewing Statistics for the SRC-ACP Directory on page 288
- Viewing Device Statistics for SRC-ACP on page 289

Viewing General Statistics for SRC-ACP

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.

Monitor Configure Diagnose Manage Logged in as: admin Refresh Preferences About Logout

ACP ACP

CLI Statistics / General

Component

Date

Disk

Interfaces...

Iptables...

JPS

NIC

NTP

Redirect Server

Route...

SAE

Security

System

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.

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.

Viewing Statistics for the SRC-ACP Directory

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 ACP

CLI Statistics / Directory

Component

Date

Disk

Interfaces...

Iptables...

JPS

NIC

NTP

Redirect Server

Route...

SAE

Security

System

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.

2. In the Slot box, enter the number of the slot for which you want to display directory statistics.
3. Click **OK**.

The Statistics/Directory pane displays statistics for the SRC-ACP directory.

Viewing Device Statistics for SRC-ACP

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.

The screenshot shows the Juniper C-Web Interface. The top navigation bar includes 'Monitor', 'Configure', 'Diagnose', and 'Manage' tabs. The left sidebar lists various components: ACP, CLI, Date, Disk, Interfaces..., Iptables..., JPS, NIC, NTP, Redirect Server, Route..., SAE, Security, and System. The main content area is titled 'Statistics / Device'. It contains three input fields: 'Filter' (with a tooltip: 'Name of the device. Value: All or part of the device name. Default: No value'), 'Slot' (with a tooltip: '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'), and 'Style' (with a tooltip: 'Output style. Choices: brief: Display only device names Default value: detail'). Below these fields are 'OK' and 'Reset' buttons. The footer shows the copyright notice for Juniper Networks, Inc. and the Juniper logo.

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.

Part 6

Using External Subscriber Monitor

- Configuring External Subscriber Events with the SRC CLI on page 293
- Monitoring External Subscriber Events with the SRC CLI on page 305
- Monitoring External Subscriber Events with the C-Web Interface on page 309

Chapter 22

Configuring External Subscriber Events 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 293
- Configuring External Subscriber Monitor on page 294
- Configuring the NIC Proxy for the Pseudo-RADIUS Server on page 297
- Configuring the Pseudo-RADIUS Server for External Subscriber Monitor on page 300
- Configuring the Client Secret for External Subscriber Monitor on page 301
- Configuring Event Notification for External Subscriber Monitor on page 302
- Starting External Subscriber Monitor on page 302
- Stopping External Subscriber Monitor on page 303

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.

For information about event notification with other third-party network devices, see *Logging In Subscribers and Creating Sessions*.

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.

Configuring External Subscriber Monitor

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 294
2. Configuring Initial Properties for External Subscriber Monitor on page 295
3. Configuring Directory Connection Properties for External Subscriber Monitor on page 295
4. Configuring Eventing Properties for External Subscriber Monitor on page 296
5. Configuring Logging Destinations for External Subscriber Monitor on page 296

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 with 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 with SRC CLI*.

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor logger file-1 file]
user@host# show
```

Configuring the NIC Proxy for the Pseudo-RADIUS Server

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 298
2. Changing the Configuration for the NIC Proxy Cache on page 298
3. Configuring a NIC Proxy for NIC Replication on page 299

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).

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).

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)*.

6. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection blacklisting]
user@host# show
```

Configuring the Pseudo-RADIUS Server for External Subscriber Monitor

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
```

Configuring the Client Secret for External Subscriber Monitor

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
```

Configuring Event Notification for External Subscriber Monitor

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
```

Starting External Subscriber Monitor

To start External Subscriber Monitor:

- Start External Subscriber Monitor from its installation directory.

```
user@host# enable component extsubmon
```

Stopping External Subscriber Monitor

To stop External Subscriber Monitor:

- Stop External Subscriber Monitor from its installation directory.

```
user@host# disable component extsubmon
```


Chapter 23

Monitoring External Subscriber Events with the SRC CLI

- Viewing Statistics for External Subscriber Monitor on page 305
- Viewing Statistics for External Subscriber Monitor Event Notifications on page 306
- Viewing Statistics for the Agent Process on page 307

Viewing Statistics for External Subscriber Monitor

Purpose View event notifications 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 15 on page 305 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 15: 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.
Number of discarded accounting requests	Number of RADIUS packets discarded.

Viewing Statistics for External Subscriber Monitor Event Notifications

Purpose View event notifications 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 16 on page 306 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 16: 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

Viewing Statistics for the Agent Process

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 17 on page 307 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 17: 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.

Chapter 24

Monitoring External Subscriber Events with the C-Web Interface

- Viewing Statistics for External Subscriber Monitor on page 309
- Viewing Statistics for External Subscriber Event Notifications on page 309
- Viewing Statistics for the Agent Process on page 309

Viewing Statistics for External Subscriber Monitor

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.

Viewing Statistics for External Subscriber Event Notifications

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.

Viewing Statistics for the Agent Process

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.

Part 7

Index

- Index on page 313

Index

A

access DNS.....165
accounting
 SAE, description.....9
ACP (Admission Control Plug-In)
 redundancy
 monitoring.....273
ACP. *See* SRC-ACP
action congestion points.....209
 configuring244
 monitoring
 C-Web interface.....282, 283
 SRC CLI.....269, 270
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.....211
APIs
 SRC-ACP.....215
APIs (application programming interfaces)
 CORBA remote API.....8
 NIC.....159
 provided with SAE.....7
 SAE core API.....8
application programming interfaces. *See* APIs
applications
 executing with SRC-ACP.....209
 external for use with SRC-ACP.....209, 212
assigned IP subscribers
 third-party devices.....81
 IP address pools.....81
assigning
 edge congestion points to subscribers.....241
 interfaces to backbone congestion point
 profiles.....251
 interfaces to subscribers.....241
ATM access network, using with SRC-ACP.....209
authorizing and tracking services.....212

B

backbone congestion point profiles
 configuring.....250

backbone congestion points.....225
 configuring.....244
 configuring for services.....245
 defining applications in.....209
 deriving.....210
 DNs of.....211
 monitoring
 SRC CLI.....268
 running applications from.....244
backbone network.....207
backbone network management with SRC-ACP
 configuring.....242
background bandwidth.....211
bandwidth
 allocating to applications not controlled by
 SRC-ACP.....211
 background.....211
 configuring
 for services.....242, 245
 for subscribers.....240
 downstream.....208
 upstream.....208
bandwidths and congestion points for subscribers
 configuring.....240
BEEP, JUNOS routing platforms.....4
 configuring port
 SRC CLI.....61
 connection.....59
Blocks Extensible Exchange Protocol. *See* BEEP

C

certificate authority (CA).....63
classification scripts
 congestion point classification
 configuring.....254
 criteria.....253, 256
 description.....253
 how it works.....253
 targets.....253, 255
Common Object Request Broker Architecture. *See*
 CORBA
community manager
 configuring, third-party devices
 SRC CLI.....88

component interactions	
JUNOS routing platforms and SAE.....	4
configuration group, JUNOS routing platforms.....	60, 70
configuration manager, instantiating for NIC.....	160
congestion point applications	
SPI for ACP.....	243
congestion point classification.....	210, 211
congestion point classification scripts. <i>See</i>	
classification scripts	
congestion point expressions.....	211, 260
congestion point profiles.....	210
congestion point expressions.....	260
defining.....	259
congestion points.....	207, 208
configuring.....	240
defining applications in.....	209
deriving.....	209
deriving from congestion point expressions.....	211
deriving from profile.....	210
managing.....	212
modifying.....	263
monitoring.....	272
retrieving information about.....	212
conventions	
notice icons.....	xxiii
text.....	xxiii
COPS (Common Open Policy Service)	
connection with JUNOSe routers.....	41
configuring SAE, SRC CLI.....	45
disabling on router.....	53
enabling on router.....	52
COPS-PR versus COPS XDR.....	3
JUNOSe router connection.....	3
CORBA (Common Object Request Broker Architecture)	
IOR location.....	155
remote API.....	8
CORBA interfaces	
SRC-ACP.....	214, 232
CORBA-based plug-in SPI. <i>See</i> plug-ins, external	
customer support.....	xxvii
contacting JTAC.....	xxvii
customized interface modules.....	8

D

deriving congestion points.....	209
device drivers	
JUNOS	
configuring, SRC CLI.....	60
viewing state, C-Web interface.....	74
viewing state, SRC CLI.....	72

viewing statistics, C-Web interface.....	75
viewing statistics, SRC CLI.....	73
JUNOSe	
configuring, SRC CLI.....	46
viewing state, C-Web interface.....	56
viewing statistics, SRC CLI.....	55
directory	
services for SRC-ACP.....	236
subscribers for SRC-ACP.....	234
directory blacklist, deleting.....	31, 36
distinguished name. <i>See</i> DN	
DN (distinguished name)	
NIC resolution.....	165
DNs	
backbone congestion points.....	211
edge congestion points.....	209
documentation set	
comments on.....	xxvii
domain maps	
reloading on SAE.....	36
downstream bandwidth.....	208

E

edge congestion points	
assigning to subscribers.....	241
deriving.....	209
DNs of.....	209
monitoring	
SRC CLI.....	266, 267
edge network.....	207, 239
edge network management, configuring.....	239
equipment registration	
deleting.....	32, 37
event notification, PCMM network	
configuration statements.....	89
properties, configuring	
SRC CLI.....	89
event notification, third-party devices	
description.....	82
events, publishing.....	225
external applications	
displaying information from.....	271
interaction with NIC.....	158
monitoring	
C-Web interface.....	286
external plug-ins	
configuring SRC-ACP as.....	207
external plug-ins. <i>See</i> plug-ins	
External Subscriber Monitor	
acting as pseudo RADIUS server, C-Web	
interface.....	293
agent process statistics, viewing	
SRC CLI.....	307
configuring.....	294
configuring basic local properties.....	294

- configuring client secret.....301
 - configuring directory connection properties.....295
 - configuring event notification.....302
 - configuring eventing properties.....296
 - configuring initial properties.....295
 - configuring logging destinations.....296
 - event notifications, viewing
 - SRC CLI.....306
 - IP address manager.....293
 - overview, C-Web interface.....293
 - starting.....302
 - statistics, viewing
 - SRC CLI.....305
 - stopping.....303
- F**
 - failover parameters, SAE.....32, 38
 - fault recovery, SRC-ACP.....214
 - files
 - ACP data.....228
- G**
 - groups, NIC hosts.....101
- H**
 - hosted internal plug-in.....225
 - hosted plug-ins. *See* plug-ins
- I**
 - interactions between SRC-ACP and other
 - components.....212, 213
 - interface classification scripts
 - reloading on SAE.....31, 36
 - interface modules, SAE.....8
 - interfaces, assigning to backbone congestion point
 - profiles.....251
 - internal plug-ins. *See* plug-ins
 - IOR
 - router initialization scripts.....48, 91
 - IP address pools
 - local address pools, configuring
 - SRC CLI.....45
 - static pools, configuring
 - SRC CLI.....45
- J**
 - JUNOS routing platforms
 - BEEP connection.....4
 - configuring port, SRC CLI.....61
 - configuration groups.....60, 70
 - configuring to interact with SAE.....68
 - default virtual router.....60
 - disabling interactions with SAE.....70
 - enabling interactions with SAE.....71
 - monitoring interactions with SAE.....71
 - SAE interactions.....4
 - SRC software process.....59
 - troubleshooting.....71
 - JUNOSe routers
 - accessing router CLI.....52
 - COPS connection.....3
 - configuring, SRC CLI.....45
 - integration overview.....41
 - monitoring interactions with SAE.....53
 - router objects, adding
 - SRC CLI.....42
 - SRC client.....41
 - starting.....52
 - stopping.....53
 - troubleshooting.....53
 - VR objects
 - adding individually, SRC CLI.....44
 - discovering, SRC CLI.....42
- L**
 - LDAP access. *See* SAE (service activation engine),
 - configuring
 - logging properties
 - configuring for SRC-ACP.....227
 - login names.....165
 - login process
 - assigned IP subscribers, third-party devices.....82
 - event notification method, third-party
 - devices.....83
 - login registration
 - deleting.....31, 37
- M**
 - managing
 - congestion points.....212
 - edge network with SRC-ACP.....239
 - manuals
 - comments on.....xxvii
 - modifying congestion points.....263
 - monitoring
 - backbone congestion points.....225
 - SRC-ACP
 - C-Web interface.....275
 - SRC CLI.....265
- N**
 - NAS port ID.....209
 - network information collector. *See* NIC
 - network interfaces.....239, 243

network publisher <i>See</i> NIC	
NIC (network information collector).....	97
API.....	159
configuration prerequisites	
C-series controllers.....	115
configuration statements.....	113
configuration, changing.....	133
configuration, verifying.....	130
data mapping.....	99
default operating properties, viewing.....	117
factory interface.....	159, 160
logging	
changing configuration.....	122
default.....	122
monitors	
example.....	197
network publisher	
overview.....	135
prerequisites.....	137
running.....	143
troubleshooting.....	145
operating properties, changing.....	117
overview.....	97
planning implementation.....	103
realms	
overview.....	165
replication	
groups.....	101
overview.....	101
SAE plug-in agents.....	128
replication, configuring.....	119
resolution processes.....	165, 166
resolvers	
constraints.....	169
overview.....	100
restarting.....	131
roles.....	166
starting.....	117
stopping.....	130
testing	
any key.....	154
examples.....	131
resolution.....	130
test data.....	154
viewing	
configuration.....	171
<i>See also</i> other NIC entries	
NIC agents	
configuration overview.....	109
directory, configuring	
SRC CLI.....	123
overview.....	100
restarting.....	132
sae client agents, configuring.....	125
sae plug-in agents, configuring.....	127
NIC configuration scenarios	
changing.....	133
SRC CLI.....	120
Multipop.....	199
OnePop.....	172
OnePopAcctId.....	185
OnePopAllRealms.....	195
OnePopDnSharedIp.....	191
OnePopDynamicIp.....	176
OnePopLogin.....	186
OnePopLoginPull.....	189
OnePopPcmm.....	174
OnePopPrimaryUser.....	189
OnePopSharedIp.....	178
OnePopStaticRouteIp.....	135, 180
OnePopVrflp.....	135, 182
overview.....	104, 171
scenario-name.....	133
NIC hosts	
configuration prerequisites.....	115
groups.....	101
overview.....	99
redundancy	
example.....	195
starting.....	117
stopping.....	131, 132
NIC locators	
external applications.....	157, 158
overview.....	97, 99
NIC proxies	
cache, configuring	
SRC CLI.....	151
configuration overview.....	147
configuration prerequisites.....	147
instantiating.....	161
logging.....	161
NIC replication, configuring	
SRC CLI.....	152
overview.....	99
prerequisites.....	148
removing instances.....	164
requirements.....	159
resolution information, configuring	
SRC CLI.....	150
resolution requests.....	162
NIC Proxy for Pseudo-RADIUS server	
configuring.....	297
NIC proxy for Pseudo-RADIUS server	
changing configuration.....	298
configuring for NIC replication.....	299
configuring resolution.....	298
NIC resolvers	
restarting.....	132
nic-network-publisher-configuration-statements.....	136
notice icons.....	xxiii

O

- operation
 - SRC-ACP, configuring.....228

P

- PacketCable Multimedia. *See* PCMM
- PCMM (PacketCable Multimedia)
 - SAE connection.....4
- plug-ins
 - architecture.....6
 - external.....6
 - hosted.....7
 - hosted internal plug-in.....225
 - internal.....6
 - SRC-ACP.....207
 - tracking
 - virtual routers, SRC CLI.....45
 - types.....5
- preventing
 - service activation.....213
- priorityList.....153
- properties
 - SRC-ACP.....227
- pseudo-RADIUS server
 - configuring External Subscriber Monitor.....300
- publishing events.....225

R

- randomPick.....153
- realm
 - See* NIC realms
- redundancy, SRC-ACP.....213
- resolution processes
 - DN to SAE reference.....192, 196, 203
 - IP address to login name.....186
 - IP address to SAE
 - reference.....172, 178, 191, 196, 200
 - login name to SAE reference.....186
- roles, NIC.....166
- roundRobin.....153
- router initialization scripts
 - iorPublisher.....48, 91
 - JUNOS
 - configuring location, SRC CLI.....67
 - JUNOSe.....48, 91
 - configuring location, SRC CLI.....51
 - example.....50, 93
 - poolPublisher.....49, 91
 - specifying for NIC.....111
- router object
 - adding for third-party devices
 - SRC CLI.....85

routers

- accessing router CLI.....52
- adding JUNOS routing platforms
 - SRC CLI.....60
- adding JUNOSe
 - SRC CLI.....42
- integrating JUNOS routing platform.....59
- integrating JUNOSe.....41

S

- SAE (service activation engine)
 - accounting.....9
 - APIs. *See* APIs
 - BEEP connection, JUNOS routing platforms.....4
 - COPS
 - JUNOSe router connection.....3
 - deleting directory blacklist.....31, 36
 - disabling interactions with JUNOS routing
 - platform.....70
 - enabling interactions with JUNOS routing
 - platform.....71
 - failover parameters.....32, 38
 - JUNOS routing platform client.....68
 - monitoring interactions
 - JUNOS routing platform.....71
 - JUNOSe routers.....53
 - NIC replication, configuring
 - SRC CLI.....128
 - overview.....3
 - PCMM environment.....4
 - plug-ins *See* plug-ins
 - reloading configuration.....30, 35
 - role.....3
 - router initialization scripts. *See* router
 - initialization scripts
 - session store
 - C-series controllers.....22
 - starting
 - SRC client on JUNOSe router.....52
 - stopping
 - SRC client on JUNOSe router.....53
- SAE (service activation engine), configuring
 - BEEP connection
 - SRC CLI.....61
 - COPS connection
 - SRC CLI.....45
 - directory eventing, SAE configuration data
 - SRC CLI.....21
 - event notification API properties
 - SRC CLI.....89
 - LDAP access, SRC CLI
 - device data.....20
 - directory data.....13
 - persistent login cache data.....19
 - policy data.....18

service data.....	16
subscriber data.....	15
router initialization script location	
SRC CLI.....	51, 67
serialized data compression	
SRC CLI.....	27
session job manager	
SRC CLI.....	28
session store	
SRC CLI.....	24
SRC-ACP.....	224
SAE (service activation engine), configuring	
to monitor backbone congestion points.....	225
SAE communities	
configuring, third-party devices	
SRC CLI.....	86
description, third-party devices.....	79
SAE remote interface	
customized interface modules.....	8
script services	
for third-party devices.....	80
serialized data compression, configuring	
SRC CLI.....	27
service activation engine. <i>See</i> SAE	
services	
configuring bandwidth for.....	242, 245
monitoring	
C-Web interface.....	278
SRC CLI.....	267
preventing activation.....	213
reloading on SAE.....	30, 35
session job manager, configuring	
SRC CLI.....	28
session store	
C-series controllers.....	22
configuring, SRC CLI	
compressing session objects.....	27
global parameters.....	26
in third-party networks.....	80
SNMP	
retrieving information from network devices.....	94
SRC Admission Control Plug-In. <i>See</i> SRC-ACP	
SRC client, JUNOS routers	
configuring.....	41
starting.....	52
stopping.....	53
SRC software process, JUNOS routing platforms.....	59
disabling.....	70
reenabling.....	71
SRC-ACP (SRC Admission Control Plug-In).....	239
API.....	215
ATM access network.....	209
authorizing and tracking services.....	212
backbone network management,	
configuring.....	242
classification scripts	
configuring.....	237
configuring.....	219
congestion points.....	207, 225
connections to services directory,	
configuring.....	236
connections to subscribers' directory,	
configuring.....	234
CORBA interfaces, configuring.....	232
data files.....	228
data files, reorganizing.....	263
description of.....	207
event publishers, configuring.....	225
external applications.....	209, 212
external plug-in for SAE, configuring.....	207
fault recovery.....	214
groups, configuring.....	220
information from external applications,	
displaying.....	271
interactions with other components.....	212
logging properties, configuring.....	227
monitoring	
C-Web interface.....	275
SRC CLI.....	265
operation, configuring.....	228
preventing service activation.....	213
properties.....	227
redundancy.....	213
configuring.....	233
SAE, configuring for.....	224
starting.....	263
state synchronization.....	214
configuring.....	233
stopping.....	263
subscribers, monitoring.....	265, 275
supporting multiple SAEs.....	212
using multiple SRC-ACPs.....	212
SRC-ACP (SRC Admission Control Plug-In), congestion	
points.....	207, 225
starting	
SRC-ACP.....	263
state synchronization	
SRC-ACP.....	214
statistics, SRC-ACP	
monitoring	
C-Web interface.....	287, 288, 289
stopping SRC-ACP.....	263
subscribers	
assigning interfaces to.....	241
configuring bandwidths and congestion points	
for.....	240
IP addresses.....	165
login names.....	165

- monitoring
 - C-Web interface.....275, 285
 - SRC CLI.....265, 271
- provisioned and actual bandwidths.....212
- subscriptions
 - reloading on SAE.....31, 36
- support, technical *See* technical support

T

- targets. *See* classification scripts
- technical support
 - contacting JTAC.....xxvii
- text conventions defined.....xxiii
- third-party devices
 - creating sessions.....81
 - integrating into SRC network
 - SRC CLI.....79
 - logging in subscribers
 - assigned IP method.....81
 - overview.....81
 - provisioning with script services.....80
 - router objects, adding
 - SRC CLI.....85
 - SAE communities.....79
 - VR objects, adding
 - SRC CLI.....87
- threads
 - configuring for sessions
 - SRC CLI.....28
- tracking plug-ins
 - virtual routers
 - SRC CLI.....45
- troubleshooting
 - JUNOS routing platforms.....71
 - JUNOSe routers.....53
- tuning factors for background bandwidth.....211

U

- upstream bandwidth.....208

V

- virtual routers
 - adding for third-party devices
 - SRC CLI.....87
 - adding individually for JUNOSe routers
 - SRC CLI.....44
 - adding operative VRs.....60
 - SRC CLI.....42

