



SRC-PE Software

Solutions Guide

Release 3.0.x

Juniper Networks, Inc.

1194 North Mathilda Avenue
Sunnyvale, California 94089
USA

408-745-2000

www.juniper.net

Part Number: 530-026634-01, 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. JUNOSe 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 Solutions Guide

Release 3.0.x

Copyright © 2008, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Writing: Linda Creed, Justine Kangas, Betty Lew

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 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

xix

Part 1	Providing Specialized Services in an SRC Environment	
Chapter 1	Managing Tiered and Premium Services with QoS on JUNOSe Routers with the SRC CLI	3
Chapter 2	Managing Subscribers for a Wireless Roaming Environment	15
Chapter 3	Configuring VoIP Services in an SRC Network	23
Chapter 4	Providing Packet Mirroring in an SRC Network	27
Part 2	Managing Services in a PCMM Environment	
Chapter 5	Providing Premium Services in a PCMM Environment	39
Chapter 6	Configuring the SAE for a PCMM Environment with the SRC CLI	55
Chapter 7	Adding Objects for CMTS Devices with the SRC CLI	67
Chapter 8	Using the NIC Resolver in a PCMM Environment	71
Chapter 9	Using PCMM Policy Servers	73
Chapter 10	Configuring the JPS with the SRC CLI	77
Chapter 11	Monitoring the JPS with the SRC CLI	101
Chapter 12	Monitoring the JPS with the C-Web Interface	105
Part 3	Managing Services on RADIUS Devices	
Chapter 13	Managing Services on Third-Party Devices in the SRC Network	115
Chapter 14	Managing Services on RADIUS-Enabled Devices	123
Chapter 15	Monitoring the Diameter Server with the SRC CLI	141
Part 4	Providing Services in IMS Networks	
Chapter 16	Providing Services in IMS Networks	147
Chapter 17	Providing Services in IMS Networks with the SRC CLI	155
Part 5	Index	
	Index	175

Table of Contents

About This Guide xix

SRC Guides and Release Notes	xix
Audience	xix
Documentation Conventions	xix
Related Juniper Networks Documentation	xxi
Obtaining Documentation	xxiii
Documentation Feedback	xxiii
Requesting Technical Support	xxiii

Part 1

Providing Specialized Services in an SRC Environment

Chapter 1

Managing Tiered and Premium Services with QoS on JUNOSe Routers with the SRC CLI 3

Overview of QoS on JUNOSe Routers	3
Dynamically Managing QoS Profiles	3
How QoS Profile Tracking Works	4
Identifying QoS Services	4
Determining the QoS Profile	5
Setting Up Policy Groups	6
Policy Group for QoS Profile Attachment Service	6
Setting Up Services	7
Reestablishing Default QoS Profile	7
Example: How QTP Activates a QoS Service	7
Configuring QoS Profile-Tracking Plug-Ins with the SRC CLI	8
Configuring Search Filters for QoS Profile-Tracking Plug-Ins	10
Updating QoS Profile Data in the Directory	12
Query Fields	12
Examples: Searching for QoS Information	13

Chapter 2

Managing Subscribers for a Wireless Roaming Environment 15

Overview of a Wireless Roaming Environment	15
Subscriber Access in a Wireless Roaming Environment	15
Configuring Subscriber Access for a Wireless Location	16
Configuring RADIUS Authentication	16
Creating Subscriber Access to an ISP	19

Creating Web Access	20
Setting Idle Timeout Options for the SAE	21

Chapter 3 Configuring VoIP Services in an SRC Network 23

Overview of Session Management for VoIP Services	23
Accounting and Tracking	23
VoIP Call Setup	24
Configuring Policies and Services for VoIP	24
Activating VoIP Services for Assigned IP Subscribers	24
Setting Timeouts for Assigned IP Subscriber Sessions	25

Chapter 4 Providing Packet Mirroring in an SRC Network 27

Overview of Packet Mirroring Services	27
Configuring Packet-Mirroring Support in an SRC Network	28
Configuring the Script Service for Packet Mirroring	28
Configuring Parameters for the Script Service for Packet Mirroring	30
Specifying Maximum Number of RADIUS Peers with the SRC CLI	32
Example: Using the Sample Packet-Mirroring Application	33
Example: Packet Mirroring for PPP Subscribers	33
Example: Packet Mirroring for DHCP Subscribers	34
Configuring DHCP Subscriber Sessions	34
Disabling RADIUS Authentication for DHCP Subscribers	34
Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API	35

Part 2 Managing Services in a PCMM Environment

Chapter 5 Providing Premium Services in a PCMM Environment 39

Overview of a PCMM Environment	39
PCMM Architecture	39
DOCSIS Protocol	40
Service Flows	41
Client Types	41
Client Type 1 Single Phase Resource Reservation Model	41
Client Type 2 Single Phase Resource Reservation Model	42
SRC Software in the PCMM Environment	43
Traffic Profiles	43
End-to-End QoS Architecture	44
Extending QoS to the Subscriber Edge Domain	45
Extending QoS to the Service Edge Domain	46
Provisioning End-to-End Services	46

	Example for Videoconferencing Services	46
	Example for Video-on-Demand Services	47
	Using the SAE in a PCMM Environment	49
	Logging In Subscribers and Creating Sessions	49
	Assigned IP Subscribers	49
	Login Interactions with Assigned IP Subscribers	50
	Event Notification from an IP Address Manager	51
	Login with Event Notification	51
	SAE Communities	52
	Storing Session Data	53
	PCMM Record-Keeping Server Plug-In	53
Chapter 6	Configuring the SAE for a PCMM Environment with the SRC CLI	55
	Configuring the SAE for a Cable Network Environment with SRC CLI	55
	Configuring the SAE to Manage PCMM Devices with SRC CLI	56
	Setting Up SAE Communities with SRC CLI	58
	Configuring the SAE Community Manager	59
	Configuring SAE Properties for the Event Notification API with SRC CLI	60
	Configuring Record-Keeping Server Peers for Plug-Ins with SRC CLI	61
	Configuring PCMM Record-Keeping Server Plug-Ins with SRC CLI	62
	Configuring CMTS-Specific RKS Plug-Ins with SRC CLI	64
Chapter 7	Adding Objects for CMTS Devices with the SRC CLI	67
	Adding Objects for CMTS Devices with the SRC CLI	67
	Creating Virtual Routers for the CMTS Device with the SRC CLI	68
Chapter 8	Using the NIC Resolver in a PCMM Environment	71
	Using the NIC Resolver in PCMM Environments	98
Chapter 9	Using PCMM Policy Servers	73
	Overview of the JPS	73
	JPS Framework	73
	JPS Interfaces	74
	Application Manager to Policy Server Interface	75
	Policy Server to RKS Interface	75
	Policy Server to CMTS Interface	75
Chapter 10	Configuring the JPS with the SRC CLI	77
	Configuration Statements for the JPS	77
	Configuring the JPS	79
	Modifying the JPS Configuration	80
	Configuring General Properties for the JPS	80

Specifying Policy Server Identifiers in Messages	81
Configuring Logging Destinations for the JPS	82
Configuring JPS to Store Log Messages in a File	82
Configuring JPS to Send Log Messages to System Logging Facility	83
Specifying Connections to the Application Managers	83
Configuring Connections to RKSs	85
Specifying Connections to RKSs	85
Configuring RKS Pairs	87
Configuring RKS Pairs for Associated Application Managers	88
Specifying Connections to CMTS Devices	89
Modifying the Subscriber Configuration	92
Configuring Subscriber IP Pools as IP Address Ranges	93
Configuring Subscriber IP Pools as IP Subnets	93
Configuring the SAE to Interact with the JPS	94
Specifying Application Managers for the Policy Server	94
Specifying Application Manager Identifiers for Policy Servers	95
Adding Objects for Policy Servers to the Directory	96
Configuring Initialization Scripts	97
Enabling State Synchronization	97
Using the NIC Resolver	98
Managing the JPS	99
Starting the JPS	99
Restarting the JPS	99
Stopping the JPS	99
Displaying JPS Status	99

Chapter 11**Monitoring the JPS with the SRC CLI****101**

Monitoring the JPS	101
Viewing Server Process Information	101
Viewing JPS State	102
Viewing Performance Statistics for the JPS Interfaces	102
Viewing Network Connections for the Application Manager	102
Viewing Network Connections for the CMTS Device	102
Viewing Performance Statistics for the CMTS Locator	103
Viewing Message Handler Information	103

Chapter 12**Monitoring the JPS with the C-Web Interface****105**

Viewing Information About the JPS Server Process with the C-Web Interface	105
Viewing JPS AM Statistics with the C-Web Interface	106
Viewing JPS AM Connections with the C-Web Interface	107
Viewing JPS CMTS Statistics with the C-Web Interface	107
Viewing JPS CMTS Connections with the C-Web Interface	108
Viewing JPS CMTS Locator Statistics with the C-Web Interface	109
Viewing JPS Message Handler Statistics with the C-Web Interface	109
Viewing JPS Message Flow Statistics with the C-Web Interface	110
Viewing JPS RKS Statistics with the C-Web Interface	111

Part 3**Managing Services on RADIUS Devices****Chapter 13****Managing Services on Third-Party Devices in the SRC Network 115**

Overview of CoA Script Service	115
Configuring CoA Script Services	115
Configuring Monitoring Agent to Receive RADIUS Accounting Messages	116
Creating the CoA Script Service with the SRC CLI	116
Configuring the CoA Script Service with the SRC CLI	117
Parameters for Sample CoA Script Service	118
Configuring Subscriptions to the CoA Script Service	119
Example: Using the Sample CoA Script Service	119
Defining RADIUS Attributes for CoA Requests with the API	120

Chapter 14**Managing Services on RADIUS-Enabled Devices 123**

Overview of the IMS AAA Server Integration	123
Managing Dynamic Services	124
Configuring the IMS AAA Server	124
Configuring the Diameter Application (SRC CLI)	125
Configuring the Diameter Application Properties	125
Configuring the Diameter Client Properties	128
Configuring the Diameter Server Properties	128
Configuring Logging Destinations	129
Configuring the NAS Groups (SRC CLI)	130
Configuring NAS Groups	130
Configuring Diameter Peers	131
Classifying Interfaces	132
Selecting Routes	133
Configuring the SAE to Manage AAA Devices	135
Configuring AAA Policies (SRC CLI)	137
Configuring AAA Policy Lists	137
Configuring AAA Policy Rules	137
Configuring Template Activation Actions	138

Chapter 15**Monitoring the Diameter Server with the SRC CLI 141**

SRC CLI Commands to Monitor the Diameter Server	141
Viewing Statistics for the Diameter Server (SRC CLI)	142
Viewing Message Handler Information for the Diameter Server (SRC CLI)	142
Viewing Server Process Information for the Diameter Server (SRC CLI)	142
Viewing Information About Diameter Server Requests (SRC CLI)	142
Viewing Diameter Server State (SRC CLI)	142

Part 4**Providing Services in IMS Networks**

Chapter 16**Providing Services in IMS Networks 147**

Overview of an IMS Environment	147
IMS and ETSI References	148
Abbreviations	148
IMS Layers	149
Signaling Protocol	150
ETSI-TISpan Architecture	150
RACS Layer	151
Rq Interface	151
SPDF	151
A-RACF	152
SRC Software in the ETSI-TISpan Architecture	152
SRC Software in the IMS Environment	153

Chapter 17**Providing Services in IMS Networks with the SRC CLI 155**

Configuration Statements for IMS Support	155
Configuring the IMS Software	156
Configuring Initial Properties for IMS	157
Configuring Directory Connection Properties for IMS	158
Configuring Initial Directory Eventing Properties for IMS	158
Configuring the Local Diameter Peer	159
Configuring the Remote Diameter Peer	160
Configuring Logging Destinations to Store Messages in a File	161
Configuring Logging Destinations to Send Messages to the System Logging Facility	162
Configuring the Subscriber Type	162
Configuring a NIC Proxy for IMS	163
Configuring Resolution Information for a NIC Proxy	163
Changing the Configuration for the NIC Proxy Cache	165
Configuring a NIC Proxy for NIC Replication	166
Configuring NIC Test Data	168
Managing IMS	169
Starting the IMS Process	169
Restarting the IMS Process	169
Stopping the IMS Process	169
Displaying IMS Status	170
Monitoring IMS with the SRC CLI	170
Viewing Server Process Information	170
Viewing Statistics for the Rq Interface	170
Monitoring IMS with the C-Web Interface	171
Viewing Statistics for the Server Process	171
Viewing Statistics for the A-RACF Rq Interface	171
Example: Configuring JUNOS Policies for IMS with the SRC CLI	172
Enabling Expansion of JUNOS Classify-Traffic Conditions	172

Part 5**Index**

Index	175
-------------	-----

List of Figures

Figure 1: Searching for All QoS Profiles on a Router	13
Figure 2: Searching for QoS Profiles in a Policy Group	14
Figure 3: Searching for All Policy Groups on a Router	14
Figure 4: Subscriber Access to a Wireless Roaming Group	16
Figure 5: PCMM Architectural Framework	40
Figure 6: Client Type 1 Single-Phase Resource Reservation Model	42
Figure 7: Client Type 2 Single-Phase Resource Reservation Model	43
Figure 8: SRC Software in the PCMM Environment	43
Figure 9: End-to-End QoS Architecture in a Cable Network	45
Figure 10: Videoconferencing Example	47
Figure 11: Video-on-Demand Example	48
Figure 12: Login Interactions with Assigned IP Subscribers	50
Figure 13: Login Interactions with Event Notification Application	51
Figure 14: SAE Community	53
Figure 15: PCMM Architectural Framework	74
Figure 16: A Simplified IMS Converged Network (Service Focus)	148
Figure 17: High-Level View of the IMS Architecture	150
Figure 18: High-Level View of the ETSI-TISPAN Architecture	151
Figure 19: SRC Software in the ETSI-TISPAN Architecture	153
Figure 20: Juniper Networks IMS Architecture	154

List of Tables

Table 1: Notice Icons	xx
Table 2: Text Conventions	xx
Table 3: Juniper Networks C-series and SRC Technical Publications	xxi
Table 4: Examples of Concatenated QoS Profile Input Values	5
Table 5: Settings for Filter Strings	11
Table 6: Packet Types for RADIUS Attributes	19
Table 7: Parameter Substitutions for Packet-Mirroring Services	30
Table 8: Parameter Substitutions for CoA Services	118
Table 9: Commands to Monitor the Diameter Server	141
Table 10: Abbreviations in the IMS and ETSI-TISPAN Environments	149

About This Guide

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

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 xx defines the notice icons used in this guide. Table 2 on page xx 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 xxi.

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 JUNOS 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 JUNOS 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 JUNOS 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

Providing Specialized Services in an SRC Environment

- Managing Tiered and Premium Services with QoS on JUNOSe Routers with the SRC CLI on page 3
- Managing Subscribers for a Wireless Roaming Environment on page 15
- Configuring VoIP Services in an SRC Network on page 23
- Providing Packet Mirroring in an SRC Network on page 27

Chapter 1

Managing Tiered and Premium Services with QoS on JUNOSe Routers with the SRC CLI

- Overview of QoS on JUNOSe Routers on page 3
- Dynamically Managing QoS Profiles on page 3
- Configuring QoS Profile-Tracking Plug-Ins with the SRC CLI on page 8
- Configuring Search Filters for QoS Profile-Tracking Plug-Ins on page 10
- Updating QoS Profile Data in the Directory on page 12
- Examples: Searching for QoS Information on page 13

Overview of QoS on JUNOSe Routers

Tiered Internet access and premium services such as video on demand, gaming, or videoconferencing require QoS profiles to be running on the subscriber interface on the JUNOSe router. The router allows only one QoS profile to be attached to an interface at one time. Therefore, as a subscriber activates and deactivates different services, the QoS profile running on the interface needs to change. Also, as subscribers activate services, they may have multiple QoS services running at the same time; for example, internet-gold with videoconferencing.

With the SRC software, you can:

- Dynamically manage QoS profiles on the JUNOSe router to control a combination of services that require QoS.
- Update the directory with a list of QoS profiles that are currently configured on a JUNOSe router.
- Search the directory for QoS policy information.

Dynamically Managing QoS Profiles

The SAE provides a QoS-tracking plug-in (QTP) that you can use to ensure that, as a subscriber activates and deactivates services, the required QoS profile is attached to the subscriber interface. With the QTP, the QoS profile selected is based on the activation state of an aggregation of services, not just one service.

For example, a subscriber activates a QoS service on a subscriber interface that requires a QoS profile that supports 512 best effort. The subscriber then activates a faster service (for example, 1024 best effort), as well as video on demand, and now has two QoS services running on an interface. The subscriber now needs a QoS profile to be attached to the interface that supports both video on demand and 1024 best-effort service. The QTP can determine which QoS profile the subscriber needs, and can cause the existing QoS profile to be removed from the subscriber interface and the new QoS profile to be attached to the interface.

Note that if a profile is installed on a subscriber interface and the QTP installs a new profile, the new profile is based on QoS services that are currently active. The new profile does not combine the functionality of the previous profile with the new profile. For example, if a subscriber has a default policy with QoS profile be-512 installed on the subscriber interface, and the subscriber activates a video-on-demand service, the QTP does not combine the functionality of be-512 with the profile that supports video on demand.

How QoS Profile Tracking Works

The SAE manages policies on router interfaces through service sessions. Service session configurations contain the policy that needs to be installed on an interface when a service is activated. The policy definition can include the name of a QoS profile to attach to the interface when the policy is installed.

When you set up the QTP, you create a QoS profile attachment service. The purpose of this service is to attach the required QoS profile to an interface. This service is hidden from subscribers and is under only QTP control.

Because profiles need to be changed only when QoS services are activated or deactivated, the QTP tracks services and reacts to service state changes by adjusting the QoS profile attachment as needed by deactivating and activating the QoS profile attachment service.

Subscribers who need their services managed by the QTP are subscribed to the QoS profile attachment service.

Identifying QoS Services

When you set up a service, you identify the service as a QoS service in one of the fields in the service definition. For example, you can assign a service name or category to indicate that the service is a QoS service, or you could assign the QTP instance name in the Tracking Plugin field.

When the SAE notifies the QTP that a service has been activated or deactivated, the QTP determines whether it is a QoS service by searching attributes in the service object. The QTP uses a search filter that you set up to search an attribute for the information that you assigned to the service to indicate that it is a QoS service.

For example, suppose you enter myqtp in the tracking plug-in field of QoS services to indicate that the service is a QoS service. You would set up the search filter to search tracking plug-in attributes for any service that contains myqtp:

```
(attribute.trackPlug=*myqtp*)
```

Or you might configure the category to indicate that a service is a QoS service. The following filter searches service category attributes for any entry that contains ultra, video on demand, or video telephony:

```
((serviceCategory=*ultra*)((serviceCategory=*video on
demand*)(serviceCategory=*video telephony*)))
```

To obtain a list of attribute names for the sspService object class, see the LDAP schema documentation in `SDK+AppSupport+Demos+Samples.tar.gz` file in the folder `SDK/doc/ldap` or on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/src>

Determining the QoS Profile

After the QTP determines that a service is a QoS service, it needs to obtain the name of the QoS profile for the service. The QTP generates a QoS profile name based on active QoS services as follows:

1. Obtains QoS profile input values.

The QTP obtains these values by taking the value of an attribute in the service definition. You specify which attribute that you want the QTP to use as the input value. For example, you can specify the service name, the category, or the contents of the design and graphics attribute.

2. Compiles a list of the QoS profile input values.
3. Removes duplicate values from the list.
4. Sorts the remaining list by using a case-sensitive alphanumeric comparison.
5. Concatenates the values with a separator. The default value for the separator is a hyphen (-). You can specify a different separator.

Table 4 on page 5 shows how lists of QoS profile input values are sorted and then concatenated.

Table 4: Examples of Concatenated QoS Profile Input Values

Input – QoS Profile Input Values	Output – Concatenated Name
be512, vod	be512-vod
game, be1024, vod	be1024-game-vod
be128	be128

6. Adds a prefix to the resulting name. The default prefix is qos-profile. (You can specify a different value.) The output from our examples now looks like this:

- qos-profile-be512-vod
- qos-profile-be1024-game-vod

- qos-profile-be128

The names that result from this process are the QoS profile names.

As you can see from this process, you need to design services and configure the QTP so that the resulting QoS profile names match the names of the QoS profiles configured on the JUNOS router.

Typically, a QoS designer creates a number of QoS profiles that support all the services that are expected to be used. This design results in various QoS profiles that need to be configured on each router. If a required QoS profile is not configured on the router, the hidden QoS profile attachment service cannot be activated. Services are still activated for the subscriber, but the services will not provide the expected traffic requirements. When this happens, the SAE logs the error but does not send an error message to the subscriber.

Setting Up Policy Groups

You need to create two types of policy groups in your QTP configuration. The QoS profile attachment service needs a policy group that attaches the required QoS profile to the subscriber interface when the attachment service is activated. QoS services need policy groups that classify traffic and specify the action to take on traffic that matches the classifier. (You can set up traffic classifiers to match any traffic.)

Policy Group for QoS Profile Attachment Service

The policy group for the hidden QoS profile attachment service must have an egress policy list with only one policy rule that contains a QoS profile attachment action. The QoS profile attachment action must have a variable parameter in the QoS profile field.



NOTE: The policy group for the QoS profile attachment service must contain only one egress policy list and must contain one and only one QoS profile attachment action. Otherwise, the SRC software will require a license for the hidden service.

When the profile attachment service is activated, the QTP substitutes the QoS profile attribute in the policy with the QoS profile name that it determined. The service then loads the policy.

The following example creates a policy group for the QoS profile attachment service. This policy group does not match any traffic.

1. Create a policy group called Pg-qos-attach, and add an egress policy list.
2. In the egress policy list, create a policy rule that has a classify-traffic condition that will not match any real traffic. For example, set both the source and destination addresses to 0.0.0.0/32.
3. In the egress policy list, create a policy rule that has a QoS profile attachment action with QoS profile qpName.

By default, the QTP looks for qpName as the variable parameter.

When the QTP determines the required QoS profile name, it substitutes qpName with the value that it acquired.

Setting Up Services

You need to set up a QoS profile attachment service and QoS services. Both types of services are value-added (SSP) services.

In the QoS profile attachment service, assign the policy group that you configured for the service. For example, policyGroupName = Pg-qos-attach, ou = ent, o = Policies, o = umc.

In QoS services, assign the policy group that you configured for the service.

Subscribe subscribers to the QoS profile attachment service and to the appropriate QoS services.

Reestablishing Default QoS Profile

A default QoS profile may be installed on the subscriber interface before the QTP installs QoS profiles in response to the activation of QoS services. For example, a profile may have been attached to the subscriber interface when the default policy was installed. Once QoS services are no longer active on the interface, the QTP can reestablish the QoS profile that was installed on the interface before the QTP began tracking services and installing profiles on the interface.

Example: How QTP Activates a QoS Service

The following example shows the process that QTP uses when a subscriber activates a QoS service. In this example, QoS profile input values are taken from the service name attribute. The hidden QoS profile attachment service is named svc-qos-attach. The svc-qos-attach service contains a policy that has the variable parameter qpName assigned as the QoS profile name.

1. The subscriber does not have any active services.
2. The subscriber activates service be512, which is a QoS service.
 - a. The SAE sends a Service Session Start event to the QTP.
 - b. The QTP searches an attribute in the service definition and determines that the service is a QoS service.
 - c. Using the SAE Common Object Request Broker Architecture (CORBA) remote application programming interface (API), the QTP gets a list of the subscriber's active QoS services.

The list contains only service be512 because that is the only service that the subscriber has activated.

- d. The QTP adds the default prefix to the QoS profile input value to obtain the QoS profile name. The result is:

qos-profile-be512

- e. The QTP deactivates the hidden svc-qos-attach service. Because this svc-qos-attach service was not active before, this operation does not have any effect.
 - f. The QTP activates the hidden svc-qos-attach service, and it substitutes variable parameter qpName with '\$qos-profile-be512' as the QoS profile name in the policy.
 - g. The policy loads qos-profile-be512 on the subscriber interface.
3. The subscriber activates service vod, which is a QoS service.
- a. The SAE sends a Service Session Start event to the QTP.
 - b. QTP searches attributes in active service definitions and determines that the service is a QoS service.
 - c. The QTP gets a list of the subscriber's active QoS services. The result is:
be512, vod
 - d. The QTP sorts the list and concatenates the QoS profile input values with the separator. The result is:
be512-vod
 - e. The QTP adds the default prefix to the concatenated name to obtain the QoS profile name. The result is:
qos-profile-be512-vod.
 - f. The QTP deactivates the hidden svc-qos-attach service.
 - g. The QTP activates the hidden svc-qos-attach service, and it substitutes variable parameter qpName with '\$qos-profile-be512-vod' as the QoS profile name in the policy.
 - h. The policy loads qos-profile-be512-vod.
4. The subscriber deactivates service vod.
- a. The QTP follows the same procedure as in Step 2 above and determines that the QoS profile name is qos-profile-vod.
 - b. The QTP deactivates the hidden svc-qos-attach service.
 - c. The QTP reactivates the hidden svc-qos-attach service, and it substitutes variable parameter qpName with '\$qos-profile-be512' as the QoS profile name in the policy.
 - d. The policy loads qos-profile-be512.

Configuring QoS Profile-Tracking Plug-Ins with the SRC CLI

Use the following configuration statements to configure the QoS profile tracking plug-in with the SRC CLI:

```
shared sae configuration plug-ins name name qos-profile-tracking {
  threads threads ;
  default-qos-profile default-qos-profile ;
  separator separator ;
  qos-profile-prefix qos-profile-prefix ;
  service-selection-attribute service-selection-attribute ;
  search-filter search-filter ;
  invisible-qos-service invisible-qos-service ;
  qos-profile-parameter-name qos-profile-parameter-name ;
}
```

1. From configuration mode for the QoS profile tracking plug-in.

```
user@host# edit shared sae configuration plug-ins name QosTracking
qos-profile-tracking
```

2. Configure the number of working threads that all QTP instances share when they process QTP events.

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set threads threads
```

3. Configure the name of the QoS profile that is attached to the interface when QoS services have been deactivated.

See Dynamically Managing QoS Profiles.

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set default-qos-profile default-qos-profile
```

4. Configure the character that is placed between QoS profile input values when the system concatenates the values during the process of creating QoS profile names.

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set separator separator
```

5. Configure the prefix added to the QoS service name as part of the process to determine the name of the QoS profile that needs to be attached to an interface for a particular service.

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set qos-profile-prefix qos-profile-prefix
```

6. Configure the name of the attribute in the service definition that you want the QTP to use as QoS profile input values.

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set service-selection-attribute service-selection-attribute
```

7. Configure the search filter that the SAE uses to search service objects in the directory to find QoS services.

See Configuring Search Filters for QoS Profile-Tracking Plug-Ins

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set search-filter search-filter
```

8. Configure the name of the hidden QoS profile attachment service that the QTP uses to attach QoS profiles to and remove QoS profiles from a router interface.

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set invisible-qos-service invisible-qos-service
```

9. Configure the name of the variable parameter used in the QoS profile name field in the QoS profile attachment action of the policy group that is assigned to the hidden QoS service.

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set qos-profile-parameter-name qos-profile-parameter-name
```

10. Verify your configuration.

```
[edit shared sae configuration plug-ins name QosTracking
qos-profile-tracking]
user@host# show
threads 1;
default-qos-profile ;
separator -;
qos-profile-prefix qos-profile;
service-selection-attribute serviceName;
search-filter (attribute.trackPlug=);
invisible-qos-service svc-qos-attach;
qos-profile-parameter-name qpName;
```

Configuring Search Filters for QoS Profile-Tracking Plug-Ins

The SAE uses a search filter to search service objects in the directory to find QoS services. You can set up the filter to search the values of any attribute in the service object, such as service name, category, or tracking plug-in. The search is successful when a value matches the filter.

To configure the search:

- Create a filter in a format similar to the LDAP search filter. Table 5 on page 11 lists the values that you can use for filters. Each filter string `<filter>` contains a simplified LDAP query.

Table 5: Settings for Filter Strings

Filter String	Action
()	Matches no objects
(*)	Matches all objects
List of <code><attribute> = <value></code> pairs <code><attribute></code> —Name of a property or attribute <code><ldapAttributeName></code> <code><value></code> —One of the following <ul style="list-style-type: none"> ■ * (asterisk) ■ Explicit string ■ String that contains an * Note: To define a special character (* & , ! \) in a string, precede it with the backslash symbol (\). 	<ul style="list-style-type: none"> ■ If <code><value></code> is *, checks for any value. ■ If <code><value></code> is an explicit string, checks whether any value of the property matches the string, regardless of case. ■ If <code><value></code> is a string that contains a *, checks whether any value of the property contains the string, regardless of case.
(& <filter> <filter> ...)	True if all filters match
(<filter> <filter> ...)	True if at least one filter matches
(! <filter>)	True if the filter does not match

The default is `attribute.trackPlug = ;`; note that you need to add a search value after the equal sign. For example:

- To search tracking plug-in attributes for any entry that contains qtp:

```
(attribute.trackPlug = *qtp*)
```

- To search service category attributes for any entry that contains ultra, video on demand, or video telephony:

```
(|(serviceCategory = *ultra*)|(serviceCategory = *video on demand*)(serviceCategory = *video telephony*))
```

- Related Topics**
- For information about obtaining a list of attribute names for the sspService object class, see the documentation for the LDAP schema in `SDK+AppSupport+Demos+Samples.tar.gz` file in the folder *SDK/doc/ldap* or on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/src>

Updating QoS Profile Data in the Directory

You can update the directory with a list of QoS profiles that are currently configured on a JUNOS router.

Query Fields

The following fields appear in the Query dialog box of the Policy Editor.

Condition Type

- Object to be searched.
- Value—router, QoS profile, or policy group
- Default—No value

Condition Value

- Name of the QoS profile, router, or policy group that you want to search.
- Value—Name of the router, QoS profile, or policy group. If you selected router or policy group as a condition type, you can select a name from the drop-down menu. If the condition type is QoS profile, continue selecting entries in the drop-down menu until you reach the name of a policy group.
- Default—No value

Find

- Object that you want to find. The software searches for this object on the QoS profile, router, or policy group defined in condition type and condition value.
- Value—Name of the router, QoS profile, or policy group. If you selected router or policy group as a condition type, you can select a name from the drop-down menu. If the condition type is QoS profile, continue selecting entries in the drop-down menu until you reach the name of a policy group.
- Default—No value

Supported

- Whether or not to search for the condition type that exists or does not exist on the router, QoS profile, or policy group.
- Value—Checked or unchecked

- Checked—Searches for the condition type that is on the router, QoS profile, or policy group
- Unchecked—Searches for the condition type that is not on the router, QoS profile, or policy group
- Default—No value

Examples: Searching for QoS Information

The query example in Figure 1 on page 13 searches for all QoS profiles on router chimera.

Figure 1: Searching for All QoS Profiles on a Router

The screenshot shows a window titled "Router Query". It contains several input fields and a checkbox. The "Aspect" field is set to "QoS Profile Configuration". The "Condition Type" dropdown is set to "Router". The "Condition Value" dropdown is set to "chimera". The "Find" dropdown is set to "QoS Profile". The "Supported" checkbox is checked. Below these fields is a text area containing the following text:

```
The following QoS Profiles are supported by Router "chimera" for QoS Profile configuration:  
aaqp  
aaqp1  
atm-default  
ethernet-default  
serial-default  
server-default
```

At the bottom of the window are three buttons: "Query", "Clear", and "Close".

The query in Figure 2 on page 14 searches for QoS profiles in policy group DHCP.

Figure 2: Searching for QoS Profiles in a Policy Group

The Router Query dialog box is shown with the following settings:

- Aspect: QoS Profile Configuration
- Condition Type: Policy Group
- Condition Value: DHCP
- Find: QoS Profile
- Supported: ☒

The results pane displays the following text:

```
The following QoS Profile is supported by Policy Group "DHCP" for QoS Profile Configuration:
atm-default atm-vc atm-vp
```

Buttons at the bottom: Query, Clear, Close.

The query in Figure 3 on page 14 searches for all policy groups that router bigfoot supports. For a policy group to be supported on a router, both the policy group and the router must contain the same QoS profile.

Figure 3: Searching for All Policy Groups on a Router

The Router Query dialog box is shown with the following settings:

- Aspect: QoS Profile Configuration
- Condition Type: Router
- Condition Value: bigfoot
- Find: Policy Group
- Supported: ☒

The results pane displays the following text:

```
The following Policy Groups are supported by Router "bigfoot" for QoS Profile configuration:
content-provider (policyGroupName=content-provider,o=Policies,o=UMC)
content-provider-fast (policyGroupName=content-provider-fast,o=Policies,o=UMC)
content-provider-medium (policyGroupName=content-provider-medium,o=Policies,o=UMC)
content-provider-slow (policyGroupName=content-provider-slow,o=Policies,o=UMC)
DHCP (policyGroupName=DHCP,o=Policies,o=UMC)
eglimit (policyGroupName=eglimit,ou=ent,o=Policies,O=UMC)
EntDefault (policyGroupName=EntDefault,ou=ent,o=Policies,O=UMC)
internet-fast (policyGroupName=internet-fast,o=Policies,o=UMC)
internet-medium (policyGroupName=internet-medium,o=Policies,o=UMC)
internet-slow (policyGroupName=internet-slow,o=Policies,o=UMC)
ISP (policyGroupName=ISP,o=Policies,o=UMC)
PPP (policyGroupName=PPP,o=Policies,o=UMC)
PPP-special (policyGroupName=PPP-special,o=Policies,o=UMC)
redirect (policyGroupName=redirect,ou=ent,o=Policies,O=UMC)
```

Buttons at the bottom: Query, Clear, Close.

Chapter 2

Managing Subscribers for a Wireless Roaming Environment

- Overview of a Wireless Roaming Environment on page 15
- Subscriber Access in a Wireless Roaming Environment on page 15
- Configuring Subscriber Access for a Wireless Location on page 16

Overview of a Wireless Roaming Environment

In a roaming wireless environment, subscribers can log in to a wireless access point at a variety of wireless locations owned by service providers that participate in a roaming network agreement. The wireless locations participating in the agreement can be owned by one or more service providers.

Typically, RADIUS manages information about subscribers between the wireless locations. A RADIUS server for an Internet service provider (ISP) manages authentication for its subscribers, and shares information with the other ISPs with which the service provider has a roaming agreement. Subscribers can log in to an SAE from any supported site.

The SAE provides support for RADIUS vendor-specific attributes for wireless Internet service provider roaming (WISPr).

- Related Topics**
- For more information RADIUS vendor-specific attributes for wireless Internet service provider roaming (WISPr): <http://www.wi-fi-alliance.org/opensection/wispr.asp>

Subscriber Access in a Wireless Roaming Environment

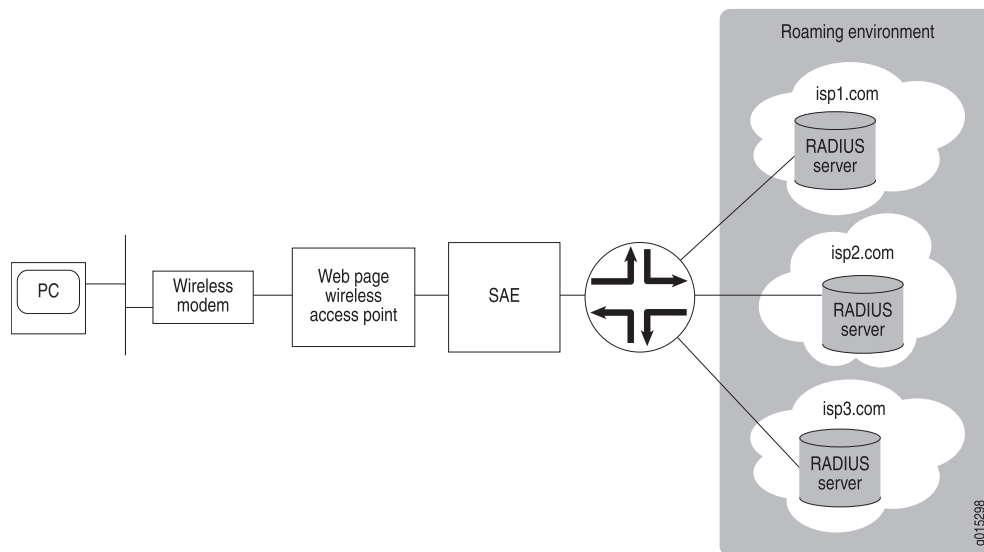
When subscribers log in to a wireless location that has a roaming agreement with other locations, the following sequence of events occurs:

1. Subscribers connect to the local wireless location and provide login information on a portal page that provides a universal access method. This login information is forwarded to the SAE.
2. Based on the login information, an access service starts.
3. The subscriber is authenticated by RADIUS; the authorization includes RADIUS vendor-specific attributes for WISPr.

4. Policies are activated for the subscriber on the router.
5. After successful start of the access service, the portal page redirects the subscriber to a specified start page.

Figure 4 on page 16 shows how subscribers interact with an SAE-managed wireless location that has a roaming agreement with wireless locations.

Figure 4: Subscriber Access to a Wireless Roaming Group



Configuring Subscriber Access for a Wireless Location

Tasks to use the SAE to manage a wireless access point that participates in a roaming agreement are:

1. Configuring RADIUS Authentication on page 16
2. Creating Subscriber Access to an ISP on page 19
3. Creating Web Access on page 20
4. Setting Idle Timeout Options for the SAE on page 21

Configuring RADIUS Authentication

You configure RADIUS authentication for users who connect from a wireless location, and set up RADIUS authentication to support a roaming environment between wireless Internet service providers. You can use the Flexible RADIUS Authentication plug-in that is provided with the SRC software, or you can create a custom RADIUS authentication plug-in.

Configuring a Custom RADIUS Authentication Plug-In

If you create a custom plug-in, be sure that it supports the same RADIUS attributes as those configured for the flexible RADIUS authentication plug-in. See “Configuring the Flexible RADIUS Authentication Plug-In” on page 17 .

For information about creating a custom plug-in, see *SAE CORBA Plug-In Service Provider Interface (SPI)* on the Juniper Networks Web site at:

<http://www.juniper.net/techpubs/software/management/src/api-index.html>

Configuring the Flexible RADIUS Authentication Plug-In

The default flexible RADIUS authentication plug-in, `flexRadiusAuth`, provides support for RADIUS vendor-specific attributes for WISPr, which are listed in the following procedure. These attributes use the IANA private enterprise number 14122 assigned to the Wi-Fi Alliance. For more information about these attributes, see <http://www.wi-fi-alliance.org/opensection/wispr.asp>

You should be familiar with the general procedure for configuring the flexible RADIUS authentication plug-in before configuring it to include the WISPr attributes. For information about configuring the flexible RADIUS authentication plug-in, see *Configuring Tracking Plug-Ins*.

When you configure the plug-in, you can use the following standard attribute values to set values in authentication response packets:

- `setAcctInterimTime`
- `SetSubstitution`
- `SetTerminateTime`

Examples in the following procedure show how you can use these attribute values.

To configure the plug-in to support a roaming environment:

1. Configure attributes.

■ Required attributes:

- An identifier for the wireless location:

`vendor-specific.WISPr.Location-ID=Identifier`

This attribute can be an interface description (`ifAlias`) or other value that identifies the JUNOS interface to which the wireless access point connects.

- The URL of the start page returned by the RADIUS server of the ISP:

`vendor-specific.WISPr.Redirection-URL=Command to make the URL available to the SRC software`

For example:

`vendor-specific.WISPr.Redirection-URL=setProperty("startURL=%s" % ATTR)`

The default configuration sets a session property named `startURL`.

- The URL of a page that a subscriber can use to log out of the network:

`vendor-specific.WISPr.Logoff-URL=URL of a log out page`

- Bandwidth attributes (recommended):

- The maximum transmission rate in bits per second:

`vendor-specific.WISPr.Bandwidth-Max-Up=Command to make the rate available to the SRC software`

For example:

`vendor-specific.WISPr.Bandwidth-Max-Up=setSubstitution("max_up_rate=%s" % ATTR)`

- The maximum receive rate in bits per second:

`vendor-specific.WISPr.Bandwidth-Max-Down=Command to make the rate available to the SRC software`

For example:

`vendor-specific.WISPr.Bandwidth-Max-Down=setSubstitution("max_down_rate=%s" % \ ATTR)`

- Optional attributes:

- The name of the wireless location:

`vendor-specific.WISPr.Location-Name=Name of the wireless location`

- The date and time that the subscriber session is to end:

`vendor-specific.WISPr.Session-Terminate-Time=Command to set the session terminate time`

For example:

`vendor-specific.WISPr.Session-Terminate-Time=setTerminateTime(ATTR)`

- The end of the subscriber session at the end of the billing day:

`vendor-specific.WISPr.Session-Terminate-End-Of-Day=ATTR or setTerminateTime("00:00:00")`

If the operator of the wireless location does not support daily billing, do not configure this attribute, and remove it if present.

- A service type for billing:

`vendor-specific.WISPr.Billing-Class-Of-Service=Service type`

2. For each attribute that you configure, configure the packet type to which the attribute applies. Table 6 on page 19 shows the packet types associated with each attribute.

Table 6: Packet Types for RADIUS Attributes

RADIUS Attribute	Associated RADIUS Packet Definition
vendor-specific.WISPr.Location-ID	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Location-ID
vendor-specific.WISPr.Redirection-URL	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Redirection-URL
vendor-specific.WISPr.Logoff-URL	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Logoff-URL
vendor-specific.WISPr.Bandwidth-Max-Up	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Bandwidth-Max-Up
vendor-specific.WISPr.Maximum-Max-Down	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Maximum-Max-Down
vendor-specific.WISPr.Location-Name	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Location-Name
vendor-specific.WISPr.Session-Terminate-Time	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Session-Terminate-Time
vendor-specific.WISPr.Session-Terminate-End-Of-Day	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Session-Terminate-End-Of-Day
vendor-specific.WISPr.Billing-Class-Of-Service	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Billing-Class-Of-Service

Creating Subscriber Access to an ISP

Configure a service that lets subscribers connect to an ISP through a captive portal, a single Web page to which subscribers connect. The policies associated with the service should specify a JUNOS policing or JUNOS rate-limiting policy to set the maximum bandwidth at which:

- A subscriber can send traffic.
- A subscriber can receive traffic.

When you configure the policies, define the bandwidth values as parameters so that the policies can be applied across a number of subscribers.

To configure a service to access the ISP:

1. Create the SRC service to use RADIUS authentication.

See Adding a Normal Service (SRC CLI).

2. Create a policy group that sets the maximum bandwidth at which a subscriber can send traffic, and the maximum bandwidth at which a subscriber can receive traffic. Use parameters to set these values.

To configure policies, see:

- Configuring Policy Groups
- Configuring Global Parameters (SRC CLI)
- Configuring Local Parameters (SRC CLI)

For example, you can create a policy configuration that includes:

- A local parameter named `max_up_rate` that sets the maximum rate at which the subscriber can send data
- A local parameter named `max_down_rate` that sets the maximum rate at which the subscriber can receive data
- A policy group `Receive(Downstream)` that references `max_down_rate`
- A policy group `Send(Upstream)` that references `max_up_rate`

Substitutions for these parameters can then be referenced in the RADIUS attributes:

```
vendor-specific.WISPr.Bandwidth-Max-Up=setSubstitution(" max_up_rate=%s" % ATTR)
vendor-specific.WISPr.Bandwidth-Max-Down=setSubstitution(" max_down_rate=%s"
% ATTR)
```

Creating Web Access

When subscribers connect to and log in to a wireless access point, they are directed to a single Web page that is referred to as a captive portal page. This page is part of a service selection portal. A captive portal page receives and manages redirected Web requests. The SRC Application Library provides an unsupported, demonstration application for a residential service selection portal.

When creating a captive portal page for a wireless roaming environment, configure the page to:

- Start an access service that is configured to be authenticated by the RADIUS server of the ISP.
- After the access service starts, redirect the subscriber to the page specified by the `Redirect-URL` RADIUS attribute. This page is the start page for the subscriber's home ISP.

You can retrieve the URL of the start page from the service session property `startURL`. Note that `startURL` is the default name used for the flexible RADIUS authentication plug-in; you can assign a different name to this property.

You can use the `Subscriber.readSubscription()` method in the Common Object Request Broker Architecture (CORBA) remote application programming interface (API) to retrieve the redirect URL.

Note that when you develop the portal, you can use the following methods in the SAE CORBA remote API to retrieve session data after the access service starts:

- `Subscriber.readSubscriber()`
- `Subscriber.readSubscription()`

For more information about these methods, see the SAE CORBA remote API documentation on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/src/api-index.html>.

Setting Idle Timeout Options for the SAE

You can configure the following options to ensure that the timeout values are consistent with the requirements for your environment:

- Idle timeout—Defines how long a session is idle before the connection is closed.
- Adjust session time—Adjusts the session time reported in an accounting message by subtracting idle time from the time if the session times out.

To configure the timeout settings:

1. Configure the service activation authentication through a RADIUS server to return an idle timeout. This configuration requires that the RADIUS server returns the idle timeout vendor-specific attribute (VSA).

or

Configure the idle timeout in the SRC service definition. For example:

```
[edit services global service service1]
user@host# set idle-timeout 5
```

Although an interval up to 5 minutes is typically recommended, for the SRC software, we recommend a minimum of 15 minutes.

2. Configure the `adjust-session-time` statement for the SAE to ensure that session time is accurately reported for accounting purposes. For example:

```
[edit shared sae group wireless configuration]
user@host# set idle-timeout adjust-session-time
```


Chapter 3

Configuring VoIP Services in an SRC Network

- Overview of Session Management for VoIP Services on page 23
- Configuring Policies and Services for VoIP on page 24
- Activating VoIP Services for Assigned IP Subscribers on page 24
- Setting Timeouts for Assigned IP Subscriber Sessions on page 25

Overview of Session Management for VoIP Services

When the SAE activates a service session, it authorizes the session with authorization plug-ins; it may use the admission control plug-in (ACP) to perform call admission control and allocate bandwidth; and it installs the policy required for the service on a JUNOS interface.

VoIP and multimedia service sessions are typically established in multiple phases that require changes to installed policies and authorized bandwidth while the service session remains active. To support VoIP sessions, the SAE allows changes to active service sessions. These changes include:

- Controlled bandwidth. If bandwidth demand increases, the authorization plug-in must authorize the change.
- Policy parameters. Only parameter substitution values can be changed. Policy parameters can include classifiers, such as destination address and port, and actions, such as rate-limit profiles.
- Session and idle timeouts. All attributes that can be set for initial service activation can be set for service session modifications.

Accounting and Tracking

Accounting information is preserved across service session changes. Accounting information for a complete service session includes the sum of counters for all service session segments.

When the ACP receives an interim update request, it compares the upstream and downstream bandwidth in the request with the current values. If the bandwidth has changed, ACP modifies its counters based on the difference between the current and new values.

Tracking plug-ins are informed of service session changes through an interim update message. The interim update is sent even if regular interim updates are disabled. If the controlled bandwidth changes, the interim update message contains the new bandwidth settings.

VoIP Call Setup

Initial setup of a VoIP call requires changes to bandwidth and to the endpoint address during call setup. The setup sequence for a VoIP call can follow this pattern:

1. The subscriber attempts to establish a call.
2. The gatekeeper (or Session Initiation Protocol [SIP] proxy) performs local admission control.
3. The gatekeeper allocates a Codec for the call; for example, 64 kbps.
4. The gatekeeper activates the VoIP service on the SAE with 64 kbps bandwidth and a destination address of unknown.
5. The SAE performs admission control, activates a service session, and installs policies on the router.
6. The gatekeeper negotiates call parameters with the remote endpoint.
7. The gatekeeper modifies the VoIP service with negotiated parameters; for example, 32 kbps, destination address 10.10.3.4, and UDP port 5678.
8. The SAE creates new policies that reflect changes to the traffic classifier and rate-limit profile, and then removes the existing policies from the router and installs the new policies.
9. The SAE sends interim updates to the ACP and tracking plug-ins.

Configuring Policies and Services for VoIP

When you set up a service that supports VoIP, you need to create a policy group for the VoIP service and assign the policy group to the VoIP service.

The SAE installs the policy on the router when the service is activated. When the service session is modified during VoIP call setup, the SAE replaces policy values with new values that were negotiated during call setup. The SAE then creates a new policy and installs it on the router.

When you set up a policy group for VoIP services, you need to assign variable parameters to fields that the SAE will need to modify. For example, source and destination addresses and UDP ports might be replaced with actual values. Upstream and downstream rate-limit parameters, such as committed rate and burst sizes, are likely to be modified.

Activating VoIP Services for Assigned IP Subscribers

When the SAE activates VoIP services, signaling proxies must identify subscriber equipment based on the IP address of the equipment. In the enterprise model, an IT manager typically subscribes to a service at a particular level in the subscriber

hierarchy, and then provides the service to all access lines and subscribers who are at lower levels in the hierarchy. In cases such as this, the SAE manages the router interface but not the subscriber. The SAE does not know the IP addresses of the subscribers and therefore cannot provide the IP address to the signaling proxies.

A type of subscriber session called assigned IP supports the case in which the SAE does not manage the subscriber but needs to provide the IP address to signaling proxies. The SAE dynamically creates an assigned IP session based on an API call. The VoIP gateway must provide the following information to the SAE before the SAE can create the assigned IP session:

- The subscriber's IP address
- The name of a managed interface (The SAE applies policies for service sessions to this interface.)
- The name of the virtual router in which the managed interface resides

The NIC maps the subscriber's IP address to the SAE reference of the managing SAE, the interface name, and the virtual router name and provides this information to the VoIP gateway.

The network information collector (NIC) keeps track of managed interfaces through a NIC SAE plug-in agent. When an interface start, stop, or interim update event occurs, the SAE sends the interface tracking events to the NIC SAE plug-in agent. The NIC uses this information as part of the process of creating these mappings.

Related Topics ■ Configuring the NIC (SRC CLI)

Setting Timeouts for Assigned IP Subscriber Sessions

To set timeouts for assigned IP subscriber sessions in the SAE configuration:

1. From configuration mode, access the SAE configuration statement that configures subscriber sessions.

```
[edit]
user@host# edit shared sae configuration subscriber-sessions
```

2. Specify the interval after which assigned IP subscriber sessions are deactivated if no service session is active.

```
[edit shared sae configuration subscriber-sessions]
user@host# set assigned-ip-idle-timeout assigned-ip-idle-timeout
```


Chapter 4

Providing Packet Mirroring in an SRC Network

- Overview of Packet Mirroring Services on page 27
- Configuring Packet-Mirroring Support in an SRC Network on page 28
- Configuring the Script Service for Packet Mirroring on page 28
- Configuring Parameters for the Script Service for Packet Mirroring on page 30
- Specifying Maximum Number of RADIUS Peers with the SRC CLI on page 32
- Example: Using the Sample Packet-Mirroring Application on page 33
- Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API on page 35

Overview of Packet Mirroring Services

Packet mirroring allows you to mirror subscriber traffic by configuring a script service with the SRC software that applies policies on a JUNOS router for RADIUS-based packet mirroring.

When the SAE activates a packet-mirroring service session, the session sends dynamic RADIUS requests, such as change-of-authorization (CoA) messages, to a RADIUS device such as a JUNOS router.

In RADIUS-based packet mirroring on a JUNOS router, a RADIUS administrator uses RADIUS attributes to configure packet mirroring of a particular subscriber's traffic. The router creates dynamic secure policies for the mirroring operation. The original traffic is sent to its intended destination, and the mirrored traffic is sent to an analyzer device (the mediation device). The mirroring operations are transparent to the subscriber whose traffic is being mirrored. This dynamic method uses RADIUS attributes and RADIUS vendor-specific attributes (VSAs) to identify a subscriber whose traffic is to be mirrored and to trigger the mirroring session. RADIUS-based mirroring uses dynamically created secure policies based on certain RADIUS VSAs. You attach the secure policies to the interface used by the mirrored subscriber. The packet-mirroring VSAs that the RADIUS server sends to the E-series router are MD5 salt-encrypted.

You must deploy RADIUS-based packet mirroring on JUNOS routers to monitor the subscriber traffic.

Related Topics ■ Configuring Packet-Mirroring Support in an SRC Network

- Example: Using the Sample Packet-Mirroring Application

Configuring Packet-Mirroring Support in an SRC Network

To support packet mirroring in an SRC network, configure a script service that can be activated to set up RADIUS-based packet-mirroring policies on a JUNOSe router. The script service defines the parameters needed to mirror subscriber traffic, such as the address of the subscriber or the analyzer device. This script service is activated for the subscriber whose traffic should be mirrored.

You must have preconfigured RADIUS-based packet mirroring on JUNOSe routers. The JUNOSe software provides RADIUS-based packet mirroring, which allows the router to create dynamic secure policies for the mirroring operation. The RADIUS administrator can configure and manage interface mirroring services that are activated by means of CoA.

To set up the SRC software for packet mirroring:

- Create a script service for packet mirroring.

The SRC software includes a sample script service that you can configure to send dynamic RADIUS requests to the JUNOSe router. You can use the sample service definition and customize it for your environment by modifying the service substitutions.

See [Configuring Parameters for the Script Service for Packet Mirroring](#).

- Configure subscriptions to the packet-mirroring service.

You can set up the subscriptions to activate immediately on login.

See [Configuring Subscriptions \(SRC CLI\)](#).

- (Optional) Configure the maximum number of RADIUS peers.

See [Specifying Maximum Number of RADIUS Peers with the SRC CLI](#).

Related Topics

- Overview of Packet Mirroring Services
- Example: Using the Sample Packet-Mirroring Application
- For information about configuring RADIUS-based packet mirroring on the JUNOSe router, see the *JUNOSe Policy Management Configuration Guide*
- For information about dynamic RADIUS requests, see RFC 3576—Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS) (July 2003)

Configuring the Script Service for Packet Mirroring

To configure the script service for packet mirroring:

1. Create a script service in the services global service *name* hierarchy or the services scope *name* service *name* hierarchy. For example:

```
[edit]
user@host# edit services global service packetMirroring
```

2. Set the type to script.

```
[edit services global service packetMirroring]
user@host# set type script
```

3. (Optional) Configure other properties as needed for your service.
4. Configure the script properties.

- a. Access the script hierarchy for the configured script service.

```
[edit services global service packetMirroring]
user@host# edit script
```

- b. Specify URL as the script type.

```
[edit services global service packetMirroring script]
user@host# set script-type url
```

- c. Specify the name of the Java class that implements the script service.

```
[edit services global service packetMirroring script]
user@host# set class-name net.juniper.smgmt.sae.packetMirroring.LiService
```

- d. Configure the URL of the script service or the path and filename of the service.

```
[edit services global service packetMirroring script]
user@host# set file file:///opt/UMC/sae/var/run/pm.jar
```

5. Verify the configuration.

```
[edit services global service packetMirroring script]
user@host# show
type script;
status active;
available;
script {
  script-type url;
  class-name net.juniper.smgmt.sae.packetMirroring.LiService;
  file file:///opt/UMC/sae/var/run/pm.jar;
}
```

6. Configure the parameters for the script service.

See Configuring Parameters for the Script Service for Packet Mirroring.

- Related Topics**
- Configuring Packet-Mirroring Support in an SRC Network
 - Adding a Normal Service (SRC CLI)

Configuring Parameters for the Script Service for Packet Mirroring

Provide parameter substitutions with the values that are in the service definitions for the script service.

Table 7 on page 30 lists the parameters specified by the sample packet-mirroring script service. In most cases, you can use the sample script service without modification.

Table 7: Parameter Substitutions for Packet-Mirroring Services

Parameter Name	Description
dynAnalyzerIPAddress	RADIUS VSA that is the IP address of the analyzer device. This attribute is required.
dynAnalyzerPortNumber	RADIUS VSA that is the UDP port number of the monitoring application in the analyzer device. If specified, dynMirrorIdentifier must also be specified.
dynMirrorIdentifier	RADIUS VSA in the form of a hexadecimal string. If specified, dynAnalyzerPortNumber must also be specified.
dynClientIp	IP address of the dynamic RADIUS client.
dynClientPort	UDP port number of the dynamic RADIUS client.
dynSecret	Shared secret.
dynRetry	Number of retries for sending dynamic RADIUS packet when no RADIUS response is received. The retry interval is 3 seconds.

Table 7: Parameter Substitutions for Packet-Mirroring Services (continued)

Parameter Name	Description
dynConfig	<p>Content of dynamic RADIUS request packets in the format < action > . < radiusAttributeName > = < pluginEventAttribute > \n</p> <ul style="list-style-type: none"> ■ action—Action that is executed on packet content (attribute) <ul style="list-style-type: none"> ■ start ■ stop ■ start-stop ■ radiusAttributeName—Valid RADIUS attribute specified as follows: <ul style="list-style-type: none"> ■ Standard RADIUS attribute name or number. ■ JUNOS VSA in one of the following formats: vendor-specific.4874. < vsa# > [.salt] 26.4874. < vsa# > [.salt] where .salt indicates that the attribute is MD5 salt-encrypted in the RADIUS packet. ■ pluginEventAttribute—Valid Python expression ■ \n—New-line character included between the lines of a configuration containing multiple lines; the entire configuration must be enclosed in quotation marks <p>For example:</p> <pre>start-stop.Acct-Session-Id = ifSessionId "start-stop.Acct-Session-Id = ifSessionId\nstart.vendor-specific.4874.58.salt = 1\nstart.vendor-specific.JUNIPER.Unisphere-Med-Dev-Handle.salt=custom['dynMirrorIdentifier']\nstart.vendor-specific.JUNIPER.Unisphere-Med-Ip-Address.salt=intIp(custom['dynAnalyzerIpAddress'])\nstart.vendor-specific.JUNIPER.Unisphere-Med-Port-Number.salt = int(custom['dynAnalyzerPortNumber'])\nstop.vendor-specific.4874.58.salt = 0"</pre>

To configure substitutions for the script parameters:

1. At the hierarchy for the script service, specify substitutions for the parameters. For example:

```
[edit services global service packetMirroring]
user@host# set parameter substitution [ dynAnalyzerIpAddress=10.227.6.221
dynAnalyzerPortNumber=9100 dynMirrorIdentifier=0x0000000100000001
dynSecret=secret dynRetry=2 dynClientIp=10.227.7.111 dynClientPort=9099
"dynConfig=\"start-stop.Acct-Session-Id =
ifSessionId\\nstart.vendor-specific.JUNIPER.Unisphere-LI-Action.salt=1\\nstart.
t.vendor-specific.JUNIPER.Unisphere-Med-Dev-Handle.salt=custom['dynMirrorId
entifier']\\nstart.vendor-specific.JUNIPER.Unisphere-Med-Ip-Address.salt=intIp(c
ustom['dynAnalyzerIpAddress'])\\nstart.vendor-specific.JUNIPER.Unisphere-Me
d-Port-Number.salt =
int(custom['dynAnalyzerPortNumber'])\\nstop.vendor-specific.JUNIPER.Unisph
ere-LI-Action.salt=0\"" ]
```

2. Verify the configuration.

```
[edit services global service packetMirroring]
user@host# show
type script;
status active;
parameter {
    substitution [ dynAnalyzerIPAddress=10.227.6.221
dynAnalyzerPortNumber=9100 dynMirrorIdentifier=0x00000000100000001
dynSecret=secret dynRetry=2 dynClientIp=10.227.7.111 dynClientPort=9099
"dynConfig=\"start-stop.Acct-Session-Id =
ifSessionId\\nstart.vendor-specific.JUNIPER.Unisphere-LI-Action.salt=
1\\nstart.vendor-specific.JUNIPER.Unisphere-Med-Dev-Handle.salt=
custom['dynMirrorIdentifier']\\nstart.vendor-specific.JUNIPER.Unisphere-Med-IP-Address.salt=

intIp(custom['dynAnalyzerIPAddress'])\\nstart.vendor-specific.JUNIPER.Unisphere-Med-Port-Number.salt=
=
int(custom['dynAnalyzerPortNumber'])\\nstop.vendor-specific.JUNIPER.Unisphere-LI-Action.salt=0\"
];
}
script {
    script-type url;
    class-name net.juniper.sgmt.scriptServices.packetMirroring.LIService;
    file file:///opt/UMC/sae/lib/pm.jar;
}
```

- Related Topics**
- Configuring Packet-Mirroring Support in an SRC Network
 - Adding a Normal Service (SRC CLI)
 - Setting Parameter Values for Services (SRC CLI)
 - Customizing Service Implementations
 - Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API

Specifying Maximum Number of RADIUS Peers with the SRC CLI

The dynamic RADIUS server can maintain a certain number of peers.

To specify the maximum number of peers with the SRC CLI:

1. From configuration mode, access the SAE configuration statement that configures dynamic RADIUS options.

```
[edit]
user@host# edit shared sae configuration dynamic-radius-server
```

2. Specify the maximum number of peers maintained by the dynamic RADIUS server.

```
[edit shared sae configuration dynamic-radius-server]
user@host# set maximum-cached-peer maximum-cached-peer
```

- Related Topics**
- Overview of Packet Mirroring Services

- Configuring Packet-Mirroring Support in an SRC Network

Example: Using the Sample Packet-Mirroring Application

To use the sample packet-mirroring application:

1. Download the SRC sample applications to your system from the Juniper Networks Web site:

<http://www.juniper.net/support/csc/swdist-erx/src.html>

2. Locate the file that contains the service definition:

`/SDK/scriptServices/packetMirroring/ldif/service.ldif`

3. Import the sample service definition to the Juniper Networks Database on the C-series Controller. To load the sample data into the database, you can use an LDAP tool, such as **ldapadd**.

You can obtain **ldapadd** from the following Web site:

<http://www.openldap.org/>

To load data into the Juniper Networks database, you need the IP address of the database and the database credentials. The default bind distinguished name (DN) for the database is `cn = umcadmin, o = umc` and the password is `admin123`.

4. Copy the `/lib/pm.jar` file used by the script service to the `/opt/UMC/sae /var/run` directory on the C-series Controller.
5. Modify the service substitutions for your environment.

You can make these substitutions by defining the parameter substitutions in the `packetMirroring` service (`serviceName = packetMirroring, o = Services, o = umc`) with the SRC CLI or by passing the values through the SAE core API.

For information about parameter substitutions, see *Configuring Parameters for the Script Service for Packet Mirroring*. For information about passing the values through the SAE core API, see *Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API*.

6. Configure a subscription to the `packetMirroring` service that is activated on login.

For information about subscriptions, see *Overview of Subscriptions*.

7. If you are modifying the sample application, copy the `sae.jar` and `logger.jar` files from the `SKD/lib` directory, and add the `sae.jar` and `logger.jar` files to the classpath when you compile your application.

Example: Packet Mirroring for PPP Subscribers

When a PPP subscriber is subscribed to the packet-mirroring service, configure the service as an activate-on-login service at user connection time. After the subscriber has logged in through the SAE remote API, the packet-mirroring service can be

subscribed to the PPP subscriber and activated. When the service is activated, a CoA request is sent to the JUNOS router that includes the PPP subscriber's accounting session ID to start packet mirroring for this subscriber.

Example: Packet Mirroring for DHCP Subscribers

When a DHCP subscriber is subscribed to the packet-mirroring service, configure the service as an activate-on-login service at user connection time. After the subscriber has logged in through the SAE remote API, the packet-mirroring service can be subscribed to the DHCP subscriber and activated. When the service is activated, a CoA request is sent to the JUNOS router that includes the DHCP subscriber's IP address and virtual router name for the JUNOS router to start packet mirroring for this subscriber.

Configuring DHCP Subscriber Sessions

You can use DHCP option 82 to identify the subscriber session. For example, if you set DHCP option 82 as the user login name, an external application can use this setting to search for the subscriber session. The following subscriber classification script illustrates this example:

```

[<name=default-user-name>?loginName=<dhcp[82]subopts[1]sing>?sub?(interfaceName=<dhcp[82]subopts[1]sing>)]
loginType = " ADDR"
[<-retailerDN->??sub?(uniqueID=<-userName->)]
retailerDN != " "
& userName != " "
[<-unauthenticatedUserDn->]
loginType == "ADDR"
loginType == "AUTHADDR"

```

Disabling RADIUS Authentication for DHCP Subscribers

Packet mirroring for DHCP subscribers does not involve RADIUS authentication, so you might have to configure authentication to grant all IP subscriber management interfaces access without authentication. For example, configure the JUNOS router with the following authentication:

```
aaa authentication ip default none
```

You can still configure other subscribers to use RADIUS authentication. For example, configure the JUNOS router with the following authentication for PPP subscribers:

```
aaa authentication ppp default radius
```

- Related Topics**
- Overview of Packet Mirroring Services
 - Configuring Packet-Mirroring Support in an SRC Network

Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API

The SRC software provides two ways to define RADIUS attributes for dynamic RADIUS authorization requests:

- Service definition
- SAE core API



NOTE: Parameters set in the API override parameters set by the service definition.

To send dynamic RADIUS authorization requests with the SAE core API, the script service uses the `sendDynamicRadius` and `getRouterDynRadiusAddr` methods in the `ServiceSessionInfo` interface to provide the content of the RADIUS packet for the dynamic authorization request to the JUNOS router that is attached to the service session.

Related Topics

- Configuring Parameters for the Script Service for Packet Mirroring
- For information about 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>

- For a sample implementation, see the following file in the `SDK+AppSupport+Demos+Samples.tar.gz` file:

`SDK/scriptServices/packetMirroring/java/net/juniper/smg/scriptServices/packetMirroring/LiService.java`

Part 2

Managing Services in a PCMM Environment

- Providing Premium Services in a PCMM Environment on page 39
- Configuring the SAE for a PCMM Environment with the SRC CLI on page 55
- Adding Objects for CMTS Devices with the SRC CLI on page 67
- Using the NIC Resolver in a PCMM Environment on page 71
- Using PCMM Policy Servers on page 73
- Configuring the JPS with the SRC CLI on page 77
- Monitoring the JPS with the SRC CLI on page 101
- Monitoring the JPS with the C-Web Interface on page 105

Chapter 5

Providing Premium Services in a PCMM Environment

- Overview of a PCMM Environment on page 39
- Using the SAE in a PCMM Environment on page 49

Overview of a PCMM Environment

The PCMM specification defines a standards-based way to deliver premium quality of service (QoS)-enhanced services across the radio frequency (RF) portion of a cable network. The PCMM capabilities of the SRC software along with Juniper Networks routers provide an end-to-end solution that seamlessly links the cable operator's RF domain with IP edge and core QoS services.

Key services supported in this environment include:

- Bandwidth on demand and variable bandwidth
- QoS-enabled streaming media, including video on demand and video telephony
- Residential voice over IP (VoIP)
- Multicast audio and video applications
- Videoconferencing
- Interactive gaming
- Peer-to-peer controls and protection services

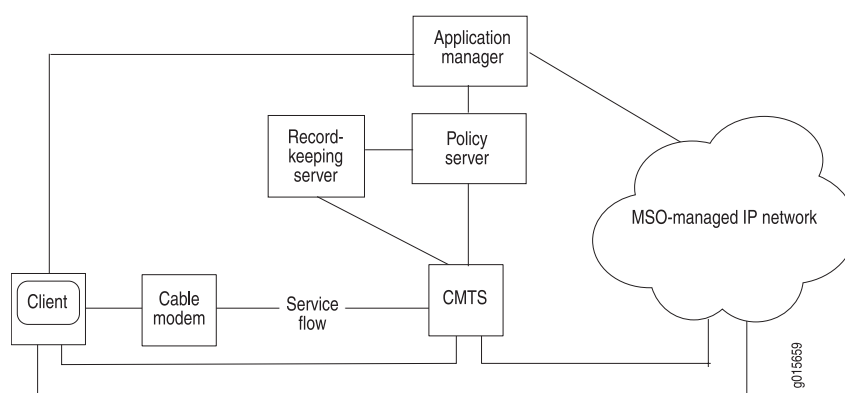
PCMM Architecture

Figure 5 on page 40 depicts the PCMM architectural framework. The basic roles of the various PCMM components are:

- Application manager—Provides an interface to policy server(s) for the purpose of requesting QoS-based service on behalf of a subscriber or a network management system. It maps session requests to resource requests and creates policies.
- Policy server—Acts as a policy decision point and policy enforcement point and manages relationships between application managers and cable modem termination system (CMTS) devices.

- CMTS device—Cable modem termination system. Performs admission control and manages network resources through Data over Cable Service Interface Specifications (DOCSIS) service flows.
- Client—Represents endpoints, such as PC applications, that can send or receive data.
- Record-keeping server—Receives event messages from other network elements, such as the policy server or CMTS device, and acts as a short-term repository for the messages. It can also assemble event messages into coherent sets or call detail records, which are then made available to other back office systems, such as billing, fraud detection, and other systems.

Figure 5: PCMM Architectural Framework



In the PCMM architecture, a client requests a multimedia service from an application manager. The application manager relays the request to a policy server. The policy server is then responsible for provisioning the policies on a CMTS device. Based on the request, the policy server records an event that indicates the policy request. The request can include network resource records, and the policy server can provide the records to a record-keeping server, such as a RADIUS accounting server.

The policy server may also provide functions such as tracking resource usage and tracking the authorization of resources on a per-subscriber, per-service, or aggregate basis.

DOCSIS Protocol

The DOCSIS protocol is the standard for providing quality of service for traffic between the cable modem and CMTS devices. The CMTS device is the headend in the DOCSIS architecture, and it controls the operations of many cable modems. Two channels carry signals between CMTS devices and cable modems:

- Downstream channels—Carry signals from the CMTS headend to cable modems.
- Upstream channels—Carry signals from the cable modems to the CMTS headend.

The DOCSIS protocol defines the physical layer and the Media Access Control (MAC) protocol layer that is used on these channels.

A cable modem usually uses one upstream channel and an associated downstream channel. Upstream channels are shared, and the CMTS device uses the MAC protocol to control the cable modem's access to the upstream channel.

Service Flows

The DOCSIS protocol uses the concept of service flows to support QoS on upstream and downstream channels. A service flow is a unidirectional flow of packets that provides a particular quality of service. Traffic is classified into a service flow, and each service flow has its own set of QoS parameters. The SRC software is compliant with the following upstream service flow scheduling types, as defined in the PacketCable Multimedia Specification PKT-SP-MM-I03-051221.

- Best effort—Used for standard Internet traffic such as Web browsing, e-mail, or instant messaging.
- Non-real-time polling service (NRTPS)—Used for standard Internet traffic that requires high throughput, and traffic that requires variable-sized data packets on a regular basis, such as high-bandwidth File Transfer Protocol (FTP).
- Real-time polling service (RTPS)—Used for applications such as Moving Pictures Experts Group (MPEG) video.
- Unsolicited grant service (UGS)—Used for real-time traffic that generates fixed-size data packets on a periodic basis. Applications include VoIP.
- Unsolicited grant service with activity detection (UGS-AD)—Used for applications such as voice activity detection, also known as silence suppression.

Downstream service flows are defined through a similar set of QoS parameters that are associated with the best-effort scheduling type on upstream service flows.

Client Types

The PCMM specification uses the concept of clients and defines a client as a logical entity that can send or receive data. The SRC software supports type 1 and type 2 clients.

The PCMM specification defines two resource reservation models for each client type—a single phase and a dual phase. The SRC software supports the single-phase model.

Client Type 1 Single Phase Resource Reservation Model

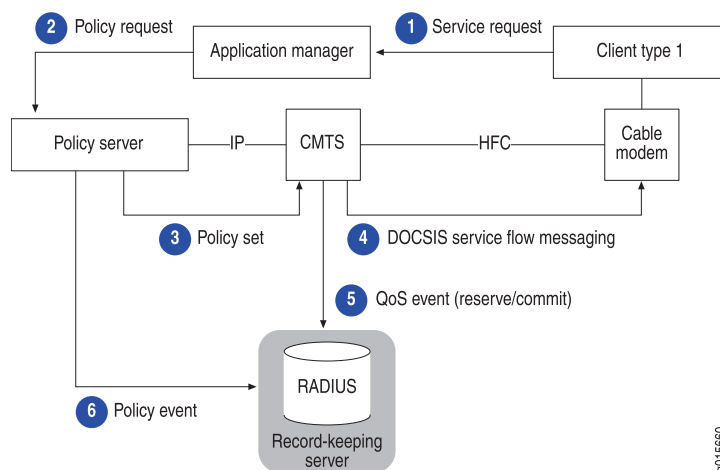
Type 1 clients represent endpoints, such as PC applications or gaming consoles, that lack specific QoS awareness or signaling capabilities. Type 1 clients communicate with an application manager to request a service. They do not request QoS resources directly from the multiple service operator (MSO) network.

Client type 1 entities support the proxied-QoS with policy-push scenario of service delivery defined in PacketCable Multimedia Architecture Framework Technical Report (PKT-TR-MM-ARCH). In this scenario, the application manager requests QoS resources on behalf of the client, and the policy server pushes the request to the CMTS device.

The CMTS device sets up and manages the DOCSIS service flow that the application requires, and might also set up and manage the cable modems.

Figure 6 on page 42 shows the message flow in an application scenario for the client type 1 single-phase resource reservation model.

Figure 6: Client Type 1 Single-Phase Resource Reservation Model

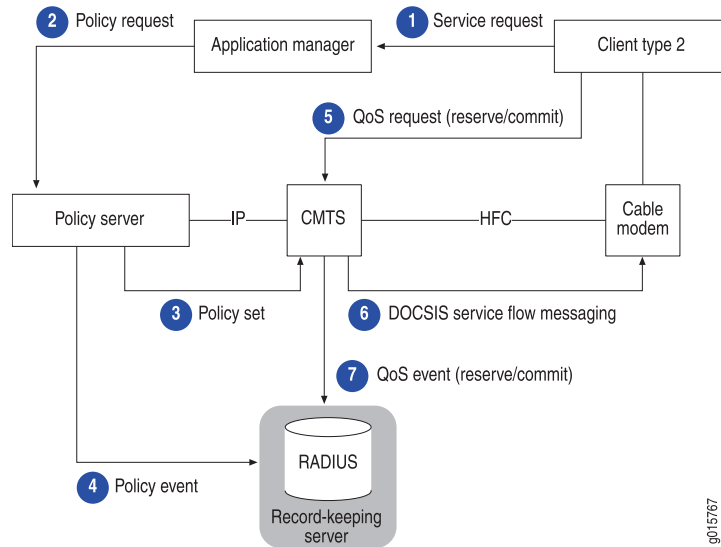


Client Type 2 Single Phase Resource Reservation Model

Type 2 clients represent endpoints that have QoS awareness or signaling capabilities. Type 2 clients communicate with an application manager to request a service and to obtain a token to present for requesting QoS resources directly from the MSO network.

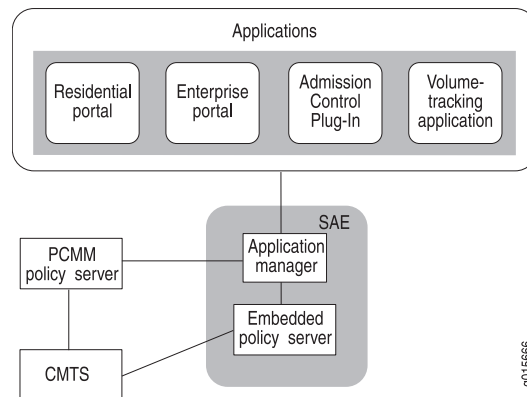
Client type 2 entities support the client-requested QoS with policy-push scenario of service delivery defined in PacketCable Multimedia Architecture Framework Technical Report (PKT-TR-MM-ARCH). In this scenario, the application manager requests QoS resources on behalf of the client, and the policy server pushes the request to the CMTS device. The CMTS device sets up and manages the DOCSIS service flow that the application requires. After the CMTS device sets up the policy, the client can request QoS resources directly from the CMTS device as long as the request is authorized by the policy server.

Figure 7 on page 43 shows the message flow in an application scenario for the client type 2 single-phase resource reservation model.

Figure 7: Client Type 2 Single-Phase Resource Reservation Model

SRC Software in the PCMM Environment

Figure 8 on page 43 shows the SRC software in the PCMM environment. The SAE is an application manager that can manage a PCMM-compliant policy server and/or a CMTS device on behalf of applications. The SAE has an embedded policy server that is not fully PCMM-compliant, but it can manage CMTS devices without requiring an external policy server. The Juniper Policy Server (JPS), a component of the SRC software that acts as a policy server, is a PCMM-compliant policy server. For more information about using the JPS, see JPS Framework .

Figure 8: SRC Software in the PCMM Environment

Traffic Profiles

The SRC software supports three types of policies that you can use to define traffic profiles between the CMTS device and the cable modem:

- DOCSIS parameters—Specifies the traffic profile through DOCSIS-specific parameters. You select the type of service flow that you want to offer, and then configure QoS parameters for the service flow.
- Service class name—Specifies the name of a service class that is configured on the CMTS device.
- FlowSpec—Defines the traffic profile through an Resource Reservation Protocol (RSVP)-like parameterization scheme. FlowSpecs support both controlled-load and guaranteed services.

You can also mark packets and then install policies that handle the marked packets in a certain way. The mark action sets the ToS byte in the IP header of IPv4 traffic or the traffic-class field in the IP header of IPv6 traffic.

For more information about traffic profiles, see *Delivering QoS Services in a Cable Environment*.

End-to-End QoS Architecture

The previous sections show how the SRC software supports QoS in the cable operator's RF domain, which encompasses the connection from the cable modem to the CMTS device. Using the SRC software along with Juniper Networks routers, you can link the RF domain to the subscriber and service edge domains.

- IP subscriber edge domain—Includes the IP network from the CMTS device to the edge router that typically connects to the cable operator's regional access network. (See "Extending QoS to the Subscriber Edge Domain" on page 45.)
- IP service edge domain—Typically includes the IP network that connects the data center that houses service delivery applications to a backbone or directly to a cable head-on facility. (See "Extending QoS to the Service Edge Domain" on page 46.)

By provisioning services across a network path, you can deliver a particular level of service for specified types of traffic. Figure 9 on page 45 shows a typical high-level architecture of a cable operator and how the SRC software and Juniper Networks routers can be deployed to deliver end-to-end QoS services.

In addition to the QoS services required in the RF domain, service policies in the subscriber edge domain that must be available for provisioning at this point include:

- Policy routing to best-of-breed appliances and premium paths
- Rate limiting, traffic shaping, and marking
- Admission control (edge resources and core resources)
- Captive portal and Web redirect capabilities
- Filtering and JUNOS routing platform-based firewall services
- JUNOS routing platform virtual private network (VPN) services

Extending QoS to the Service Edge Domain

The service edge domain includes service edge routers that aggregate applications. To support QoS in service edge domains, the SRC software sends policies to a service edge router that provides for enhanced service delivery to the service origination edge for centralized or hosted services, such as multimedia or VoD.

In addition to the QoS services required in the RF domain, service policies in the service edge domain that must be capable of being provisioned at this point include:

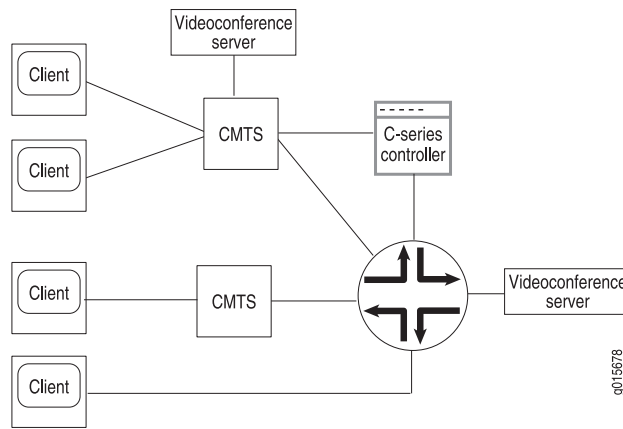
- Policy routing to best-of-breed appliances and premium paths
- Rate limiting, traffic shaping (called hierarchical queuing in JUNOS software), and marking
- Filtering and JUNOS routing platform based firewall services
- JUNOS routing platform VPN services

Provisioning End-to-End Services

The following sections provide examples of how you can use the SRC software to provision services for video applications. Although the examples show one SAE managing all the network devices, separate SAEs could manage each device and provide the same service.

Example for Videoconferencing Services

You can configure services to mark traffic forwarded from specified systems, and then apply an end-to-end service level for that traffic. Figure 10 on page 47 shows a scenario in which videoconferencing is delivered in a PCMM environment.

Figure 10: Videoconferencing Example

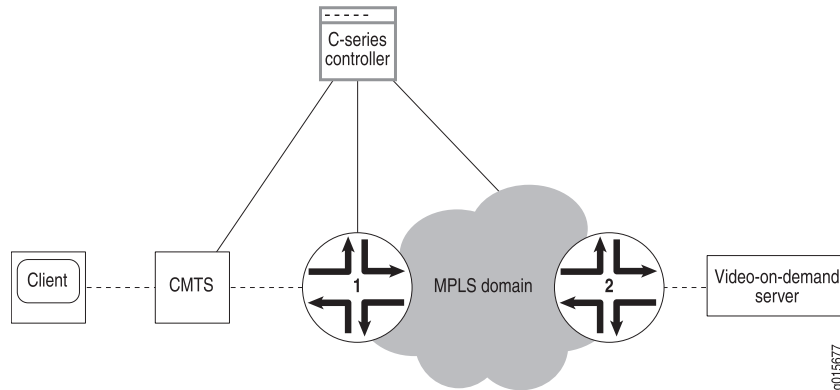
To ensure a specified level of service from each client PC to the videoconference server and then to each client PC participating in the videoconference, you could configure the following types of services:

- Three services:
 - A service that provides policies to mark packets with a specified type of service for the videoconferencing software.
 - A service that provides policies for the type of service specified for CMTS device.
 - A service that provides policies for the type of service specified for the JUNOS routing platform or JUNOSe router.
- An infrastructure service for each service.
- An aggregate service that contains the three infrastructure services as fragment services.

This configuration marks packets that the CMTS device receives from both client and server, and applies forwarding policies on the CMTS device and on the JUNOSe router or JUNOS routing platform for packets sent to and received from the videoconferencing server.

Example for Video-on-Demand Services

You can configure services to provide server-to-client service for traffic sent from a video-on-demand server to client PCs. Figure 11 on page 48 shows a scenario in which video on demand is delivered in a PCMM environment.

Figure 11: Video-on-Demand Example

To ensure a specified level of service from the video-on-demand server to the client PC, you could configure the following types of services:

- Services that provide bandwidth-on-demand (BoD) policies for traffic that is being forwarded from the video-on-demand server through:
 - JUNOS routing platforms
 - CMTS devices
- A script service that sets up the Multiprotocol Label Switching (MPLS) path and delivers the specified service level for traffic that is being forwarded from the video-on-demand server through the MPLS domain.
- An infrastructure service for each value-added and script service.
- An aggregate service that contains all the infrastructure services as fragment services.

This configuration applies BoD policies to the two JUNOS routing platforms, the MPLS domain, and the CMTS device, and sets up the MPLS path from JUNOS routing platform (2) to JUNOS routing platform (1).

- Related Topics**
- For more information about each scheduling type, see [Delivering QoS Services in a Cable Environment](#).
 - For more information about PCMM, consult the following specifications provided by CableLabs:
 - PacketCable Multimedia Architecture Framework Technical Report (PKT-TR-MM-ARCH)
 - PacketCable Multimedia Specification PKT-SP-MM-I03-051221
 - PacketCable Security Specifications (PKT-SP-SEC)

Using the SAE in a PCMM Environment

The SAE uses the Common Open Policy Service (COPS) protocol as specified in the PacketCable Multimedia Specification PKT-SP-MM-I03-051221 to manage 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.

This section describes how the SAE is used in cable networks. It includes the following topics:

- Logging In Subscribers and Creating Sessions on page 49
- SAE Communities on page 52
- Storing Session Data on page 53

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 CMTS 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 Advanced Services Gateway (ASG).

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

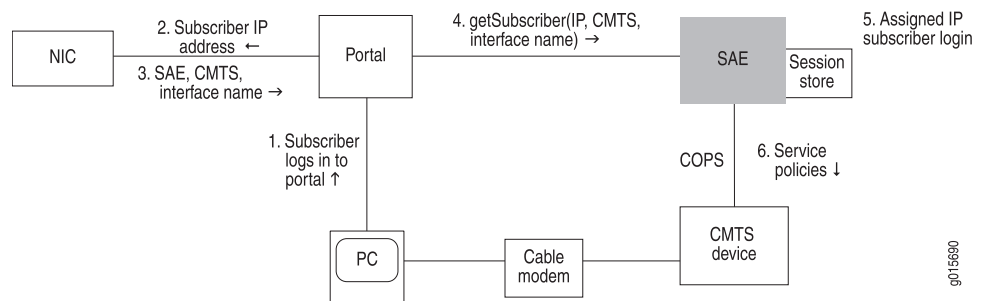
With the assigned IP subscriber method of logging in subscribers and creating sessions, the SRC software uses IP address pools along with NIC resolvers to provide

mapping of IP addresses to SAEs. You configure the static address pools or dynamically discovered address pools in the virtual router configuration for a CMTS device. These pools are published in the NIC. The NIC maps subscriber IP addresses in requests received through the portal or Advanced Services Gateway to the SAE that currently manages that CMTS device.

Login Interactions with Assigned IP Subscribers

This section describes login interactions for assigned IP subscribers. In the example shown in Figure 12 on page 50, the subscriber activates a service through a portal. You could also have the subscriber activate a service through the Advanced Services Gateway.

Figure 12: 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, CMTS 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, CMTS device, and interface name.
5. The SAE creates an assigned IP subscriber and performs a subscriber login. Specifically, it:
 - a. Runs the interface classification script and creates a pseudointerface for the PCMM device driver.
 - If it finds a default policy, it pushes the policy to the CMTS device.
 - If it does not find a default policy, it continues with the next steps.
 - b. Runs the subscriber classification script with the IP address of the subscriber. (Use the `ASSIGNEDIP` login type in subscriber classification scripts.)
 - c. Loads the subscriber profile.
 - d. Runs the subscriber authorization plug-ins.

- e. Runs the subscriber tracking plug-ins.
 - f. 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 CMTS 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

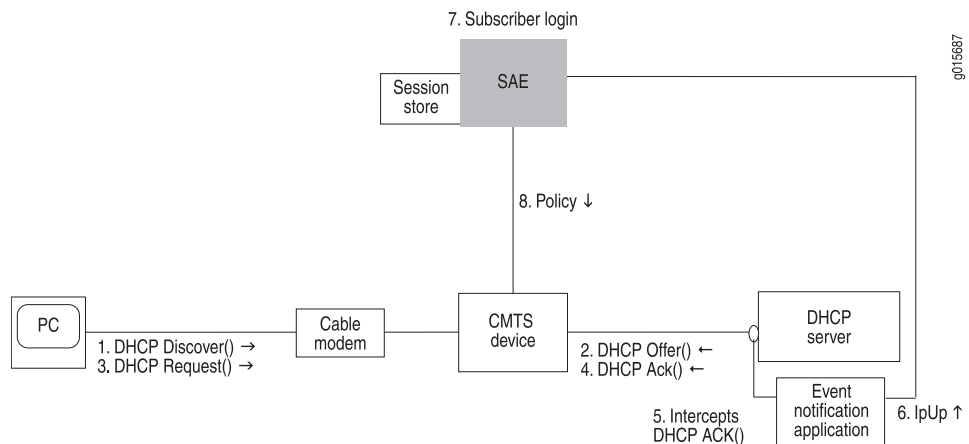
With the event notification method of logging in subscribers and creating subscriber sessions, the subscriber logs in to the CMTS device and obtains an IP address through an address server, usually a DHCP server. The SAE receives notifications about the subscriber, such as the subscriber's IP address, 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 CMTS device. You can also use Monitoring Agent, an application that was created with the event notification API, and that monitors DHCP or RADIUS messages for DHCP or RADIUS servers. See *SRC-PE Sample Applications Guide*.

Login with Event Notification

This section describes login interactions using event notifications.

Figure 13: 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.

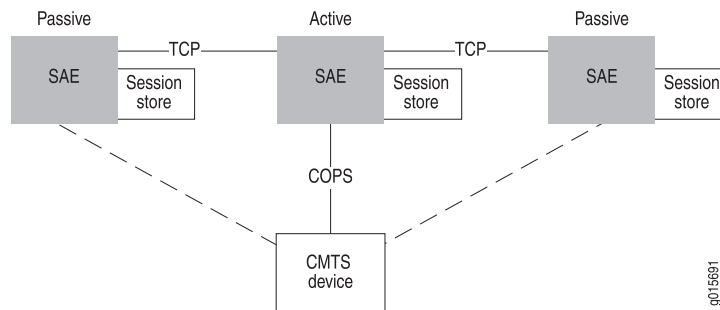
3. The DHCP client sends a DHCP request to the DHCP server.
4. The DHCP server acknowledges the request by sending a DHCP Ack message to the DHCP client.
5. The event notification application that is running on the DHCP server intercepts the DHCP Ack message.
6. The event notification application sends 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 interface classification script and creates a pseudointerface for the PCMM device driver.
 - If it finds a default policy, it pushes the policy to the CMTS device.
 - If it does not find a default policy, it continues with the next steps.
 - b. Runs the subscriber classification script.
 - c. Loads the subscriber profile.
 - d. Runs the subscriber authorization plug-ins.
 - e. Runs the subscriber tracking plug-ins.
 - f. Creates a subscriber session and stores the session in the session store file.
8. The SAE provisions policies for the subscriber session on the CMTS device.

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 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.

SAE Communities

For SAE redundancy in a cable 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 CMTS device and keeps session data up to date within the community. Figure 14 on page 53 shows a typical SAE community.

Figure 14: SAE Community

When an SAE opens a connection to the CMTS device, it negotiates with other SAEs to determine which SAE controls the CMTS 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 or the connection between the CMTS device and the active SAE goes 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 CMTS device, session data is stored locally 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 Fault Recovery.

PCMM Record-Keeping Server Plug-In

To allow the SAE's embedded policy server to communicate with a record-keeping server (RKS) in a PCMM environment, you need to use the PCMM record-keeping server plug-in. This plug-in is similar to the RADIUS accounting plug-ins, but it works with any RKS that is compliant with the PCMM specification. The RKS plug-in supports additional attributes: Application-Manager-ID, Request-Type, and Update-Reason. The plug-in sends all requests to the RKS as Acct-Status-Type = Interim-Update.

Chapter 6

Configuring the SAE for a PCMM Environment with the SRC CLI

- Configuring the SAE for a Cable Network Environment with SRC CLI on page 55
- Configuring the SAE to Manage PCMM Devices with SRC CLI on page 56
- Setting Up SAE Communities with SRC CLI on page 58
- Configuring the SAE Community Manager on page 59
- Configuring SAE Properties for the Event Notification API with SRC CLI on page 60
- Configuring Record-Keeping Server Peers for Plug-Ins with SRC CLI on page 61
- Configuring PCMM Record-Keeping Server Plug-Ins with SRC CLI on page 62
- Configuring CMTS-Specific RKS Plug-Ins with SRC CLI on page 64

Configuring the SAE for a Cable Network Environment with SRC CLI

The tasks to configure the SAE for a cable network environment are:

1. Configure the SAE to manage PCMM devices.

Configuring the SAE to Manage PCMM Devices with SRC CLI.

2. Configure the session store.

See Configuring the Session Store Feature.

3. Set up SAE communities.

See Setting Up SAE Communities with SRC CLI.

4. (Optional) Configure SAE properties for the event notification API.

See Configuring SAE Properties for the Event Notification API with SRC CLI (if you are using an external address manager).

5. (Optional) Configure record-keeping server peers for plug-ins.

See Configuring Record-Keeping Server Peers for Plug-Ins with SRC CLI (if you are using the RKS plug-in).

6. (Optional) Configure PCMM record-keeping server plug-ins.

See Configuring PCMM Record-Keeping Server Plug-Ins with SRC CLI (if you are using the SAE's embedded policy server).

7. (Optional) Configure CMTS-specific RKS plug-ins.

See Configuring CMTS-Specific RKS Plug-Ins with SRC CLI.

In addition to configuring the SAE, you need to:

1. Configure the CMTS device in the directory (if you are using the SAE's embedded policy server).

See Adding Objects for CMTS Devices with the SRC CLI .

2. Configure the NIC (if you are using assigned IP subscribers).

See Using the NIC Resolver.

3. Enable the Common Open Policy Service (COPS) interface on the CMTS device. See the documentation for your CMTS device for information about how to do this.

- Related Topics**
- To use the C-Web interface to configure the SAE for a PCCM environment, see Configuring the SAE to Manage PCMM Devices (C-Web Interface).
 - For information about setting up SAE groups, see Configuring an SAE Group .
 - For information about setting up a community manager, see Setting Up SAE Communities with SRC CLI .

Configuring the SAE to Manage PCMM Devices with SRC CLI

The SAE connects to the PCMM device by using a COPS over TCP connection. The PCMM device driver controls this connection.

Use the following configuration statements to configure the SAE to manage CMTS devices:

```
shared sae configuration driver pcmm {
    keepalive-interval keepalive-interval ;
    tcp-connection-timeout tcp-connection-timeout ;
    application-manager-id application-manager-id ;
    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 ;
    sae-community-manager sae-community-manager ;
    disable-full-sync disable-full-sync ;
    disable-pcmm-i03-policy disable-pcmm-i03-policy ;
    session-recovery-retry-interval session-recovery-retry-interval ;
    element-id element-id ;
    default-rks-plug-in default-rks-plug-in ;
}
```

To configure the SAE to manage CMTS devices:

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

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

2. Configure the interval between keepalive messages sent from the COPS client (the PCMM device) to the COPS server (the SAE).

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

3. Configure the timeout for opening a TCP connection to the PCMM device.

```
[edit shared sae group west-region configuration driver pcmm]
user@host# set tcp-connection-timeout tcp-connection-timeout
```

4. When this SAE is configured as the application manager, configure the identifier of the application manager.

```
[edit shared sae group west-region configuration driver pcmm]
user@host# set application-manager-id application-manager-id
```

5. Configure the time that the COPS server (the SAE) waits for a response to COPS requests from the COPS client (the PCMM device). Change this value only if a high number of COPS timeout events appear in the error log.

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

6. Configure the maximum length of a COPS message.

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

7. Configure the buffer size for receiving COPS messages from the COPS client.

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

8. Configure the buffer size for sending COPS messages to the COPS client.

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

9. Configure the name of the community manager that manages PCMM driver communities. Active SAEs are selected from this community.

```
[edit shared sae group west-region configuration driver pcmm]
user@host# set sae-community-manager sae-community-manager
```

10. Enable or disable state synchronization with PCMM policy servers.

```
[edit shared sae group west-region configuration driver pcmm]
user@host# set disable-full-sync disable-full-sync
```

11. Enable or disable the SAE to send classifiers to the router that comply with PCMM IO3. Disable this option if your network deployment has CMTS devices that do not support PCMM IO3.

```
[edit shared sae group west-region configuration driver pcmm]
user@host# set disable-pcmm-i03-policy disable-pcmm-i03-policy
```

12. Configure the time between attempts by the SAE to restore service sessions that are being recovered in the background when state synchronization completes with a state-data-incomplete error.

```
[edit shared sae group west-region configuration driver pcmm]
user@host# set session-recovery-retry-interval session-recovery-retry-interval
```

13. (Optional) Configure the unique identifier that the SAE uses to identify itself when it originates in record-keeping server (RKS) events.

```
[edit shared sae group west-region configuration driver pcmm]
user@host# set element-id element-id
```

14. (Optional) Specify the name of the default RKS plug-in to which the SAE sends events for CMTS devices.

```
[edit shared sae group west-region configuration driver pcmm]
user@host# set default-rks-plug-in default-rks-plug-in
```

15. (Optional) Verify your PCMM driver configuration.

```
[edit shared sae group west-region configuration driver pcmm]
user@host# show
keepalive-interval 45;
tcp-connection-timeout 5;
application-manager-id 1;
message-timeout 120000;
cops-message-maximum-length 204800;
cops-message-read-buffer-size 3000;
cops-message-write-buffer-size 3000;
sae-community-manager PcmmCommunityManager;
disable-full-sync true;
disable-pcmm-i03-policy true;
session-recovery-retry-interval 3600000;
element-id 1;
default-rks-plug-in rksTracking;
```

Setting Up SAE Communities with SRC CLI

You can configure the following for SAE communities:

- Define the members of an SAE community by adding the IP addresses of SAEs in the community to the virtual router object of the network device in the directory.

See Creating Virtual Routers for the CMTS Device with the SRC CLI .

- Configure parameters for the SAE community manager.

See Configuring the SAE Community Manager .

- Specify the name of the community manager with the `set sae-community-manager` option in the PCMM driver configuration.

See Configuring the SAE to Manage PCMM Devices with SRC CLI.

- If there is a firewall in the network, configure the firewall to allow SAE messages through.

Configuring the SAE Community Manager

Use the following configuration statements to configure the SAE community manager that manages PCMM 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, `west_region` is the name of the SAE group, and `sae_mgr` is the name of the community manager.

```
user@host# edit shared sae group west-region 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 group west-region 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 group west-region 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 group west-region 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 group west-region 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 group west-region configuration
external-interface-features sae_mgr CommunityManager]
user@host# show
CommunityManager {
  keepalive-interval 30;
  threads 5;
  acquire-timeout 15;
  blackout-time 30;
}
```

Related Topics ■ Initially Configuring the SAE .

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

Configuring Record-Keeping Server Peers for Plug-Ins with SRC CLI

An RKS peer is an instance of an RKS. A PCMM environment has a primary RKS and optionally a secondary RKS. The primary RKS is mandatory, and you assign the RKS as primary by configuring it as the default peer in the RKS plug-in. The secondary RKS is optional, and it is an RKS peer that is not configured as the default peer. If you define multiple nondefault peers, one of them is randomly chosen to be the secondary RKS.

RKS peers are configured in the peer group for each PCMM RKS plug-in instance. To create an RKS peer group:

Use the following configuration statements to configure an RKS peer group.

```
shared sae configuration plug-ins name name pcmm-rks peer-group name {
    server-address server-address ;
    server-port server-port ;
}
```

To configure an RKS peer group:

1. From configuration mode, access the configuration statements for RKS plug-ins. In this sample procedure, west-region is the name of the SAE group, and rksPlugin is the name of the plug-in and rksPeer is the name of the peer group.

```
user@host# edit shared sae group west-region configuration plug-ins name  
rksPlugin pcmm-rks peer-group rksPeer
```

2. Specify the IP address of the RKS server to which the SAE sends accounting data.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks  
peer-group rksPeer]  
user@host# set server-address server-address
```

3. Specify the port used for sending accounting packets.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks  
peer-group rksPeer]  
user@host# set server-port server-port
```

4. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin  
pcmm-rks peer-group rksPeer]  
user@host# show  
server-address 10.10.3.60;  
server-port 1812;
```

Related Topics ■ Initially Configuring the SAE .

Configuring PCMM Record-Keeping Server Plug-Ins with SRC CLI

Use the following configuration statements to configure an RKS plug-in.

```
shared sae configuration plug-ins name name pcmm-rks {  
  load-balancing-mode (failover | roundRobin);  
  fallback-timer fallback-timer;  
  retry-interval retry-interval ;  
  maximum-queue-length maximum-queue-length ;  
  bind-address bind-address ;  
  udp-port udp-port ;  
  feid-mso-data feid-mso-data ;  
  feid-mso-domain-name feid-mso-domain-name ;  
  trusted-element;  
  default-peer default-peer ;  
}
```

To configure an RKS plug-in:

1. From configuration mode, access the configuration statements for RKS plug-ins. In this sample procedure, west-region is the name of the SAE group, and rksPlugin is the name of the plug-in.

```
user@host# edit shared sae group west-region configuration plug-ins name  
rksPlugin pcmm-rks
```

2. Specify the mode for load-balancing RKSs.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]  
user@host# set load-balancing-mode (failover | roundRobin)
```

3. Specify if and when the SAE attempts to fail back to the default peer.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]  
user@host# set fallback-timer fallback-timer
```

4. Specify the time the SAE waits for a response from an RKS before it resends the packet.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]  
user@host# set retry-interval retry-interval
```

5. Specify the maximum number of unacknowledged messages that the plug-in receives from the RKS before it discards new messages.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]  
user@host# set maximum-queue-length maximum-queue-length
```

6. (Optional) Specify the source IP address that the plug-in uses to communicate with the RKS.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]  
user@host# set bind-address bind-address
```

7. (Optional) Specify the source UDP port or a pool of ports that the plug-in uses to communicate with the RKS.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]  
user@host# set udp-port udp-port
```

8. (Optional) Specify the multiple service operator (MSO)—defined data in the financial entity ID (FEID) attribute, which is included in event messages.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]  
user@host# set feid-mso-data feid-mso-data
```

9. (Optional) Specify the MSO domain name in the FEID attribute that uniquely identifies the MSO for billing and settlement purposes.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]  
user@host# set feid-mso-domain-name feid-mso-domain-name
```

10. (Optional) When the SAE is running as a policy server—which means that the SAE sends event messages directly to the RKS—enable the SAE as a trusted network element.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
user@host# set trusted-element
```

11. Specify the name of the primary RKS peer to which the SAE sends accounting packets.

See Configuring Record-Keeping Server Peers for Plug-Ins with SRC CLI.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
user@host# set default-peer default-peer
```

12. (Optional) Verify your RKS plug-in configuration.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin
pcmm-rks]
user@host> show
load-balancing-mode failover;
failback-timer -1;
retry-interval 3000;
maximum-queue-length 10000;
feid-mso-domain-name abcd.com;
trusted-element;
default-peer radius01;
```

13. (Optional) Specify an RKS plug-in for specific CMTS devices.

See Configuring CMTS-Specific RKS Plug-Ins with SRC CLI .

Related Topics ■ Initially Configuring the SAE .

Configuring CMTS-Specific RKS Plug-Ins with SRC CLI

You can configure an RKS plug-in for specific CMTS devices. When there are events for the CMTS device, the SAE sends the events to the specified plug-in.

Use the following configuration statement to assign a CMTS-specific RKS plug-in.

```
shared sae configuration driver pcmm cmts-specific-rks-plugin-ins name {
  rks-plugin-in rks-plugin-in ;
}
```

To configure a CMTS-specific RKS plug-in:

1. From configuration mode, access the configuration statements for RKS plug-ins. In this sample procedure, west-region is the name of the SAE group, and cmtsPlugin is the name of the plug-in assignment.

```
user@host# edit shared sae group west-region configuration driver pcmm
cmts-specific-rks-plugin-ins cmtsPlugin
```

2. Specify the name of the CMTS-specific RKS plug-in.

```
[edit shared sae group west-region configuration driver pcmm
 cmts-specific-rks-plugin-ins cmtsPlugin]
user@host# set rks-plugin-in rks-plugin-in
```

3. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration driver pcmm
 cmts-specific-rks-plugin-ins cmtsPlugin]
user@host# show
rks-plugin-in rksPlugin;
```

Related Topics ■ Initially Configuring the SAE .

Chapter 7

Adding Objects for CMTS Devices with the SRC CLI

- Adding Objects for CMTS Devices with the SRC CLI on page 67
- Creating Virtual Routers for the CMTS Device with the SRC CLI on page 68

Adding Objects for CMTS Devices with the SRC CLI

To manage CMTS devices, the SAE creates and manages pseudointerfaces that it associates with a virtual router object. Each CMTS device in the SRC network must appear in the configuration as a router object, and it must be associated with a virtual router object called default. The router and virtual router are not actually configured on the CMTS device; the router and virtual router provide a way for the SAE to manage the CMTS device by using the SAE's embedded policy server.

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 | proxy);  
  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. In this sample procedure, pcmm_dtr is the name of the object.

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

2. (Optional) Add a description for the CMTS device.

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

3. Add the IP address of the CMTS device.

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

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

```
[edit shared network device pcmm_dtr]
user@host# set device-type pcmm
```

5. (Optional) Verify your configuration.

```
[edit shared network device pcmm_dtr]
user@host# show
description "CMTS device";
management-address 192.168.3.5;
device-type pcmm;
interface-classifier {
  rule rule-0 {
    script #;
  }
}
```

Creating Virtual Routers for the CMTS Device with the SRC CLI

You need to add a virtual router object called default to the CMTS device.

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. In this sample procedure, `pcmm_dtr` is the name of the router and `default` is the name of the virtual router.

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

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

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

5. (Optional) Specify service scopes assigned to this virtual router.

See Configuring Service Scopes.

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

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

If you are using assigned IP subscribers along with the network information collector (NIC), you need to configure either a local or static address pool so that the NIC can resolve the IP-to-SAE mapping.

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

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

If you are using assigned IP subscribers along with the NIC, you need to configure either a local or static address pool so that the NIC can resolve the IP-to-SAE mapping.

```
[edit shared network device pcmm_dtr virtual-router default]
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 pcmm_dtr virtual-router default]
user@host# tracking-plugin [ tracking-plugin ...]
```

9. (Optional) Verify your configuration.

```
[edit shared network device pcmm_dtr virtual-router default]
user@host# show
sae-connection [ 10.14.39.2 10.10.5.30 ];
snmp-read-community *****;
snmp-write-community *****;
scope POP-Westford;
local-address-pools "10.25.8.0 10.25.20.255";
tracking-plugin rksPlugin;
```


Chapter 8

Using the NIC Resolver in a PCMM Environment

- Using the NIC Resolver in PCMM Environments on page 98

Using the NIC Resolver in PCMM Environments

If you are using the NIC to map the subscriber IP address to the SAE, you need to configure a NIC host. The NIC system uses IP address pools to map IP addresses to SAEs. You configure the local address pools in the application manager configuration for a policy server group. These pools are published in the NIC. The NIC maps subscriber IP addresses in requests received through the portal or Advanced Services Gateway to the policy server group that currently manages that CMTS device.

The OnePopPcmm sample configuration data supports this scenario for a 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 OnePopPcmm configuration supports one point of presence (POP). NIC replication can be used to provide high availability. The realm for this configuration accommodates the situation in which IP pools are configured locally on each application manager group 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 following agents collect information for resolvers in this realm:

- Directory agent PoolVr collects and publishes information about the mappings of IP pools to the policy server group.
- Directory agent VrSaeld collects and publishes information about the mappings of policy server groups to SAEs.

Related Topics ■ Specifying Application Manager Identifiers for Policy Servers (C-Web Interface)

Chapter 9

Using PCMM Policy Servers

- Overview of the JPS on page 73
- JPS Framework on page 73
- JPS Interfaces on page 74

Overview of the JPS

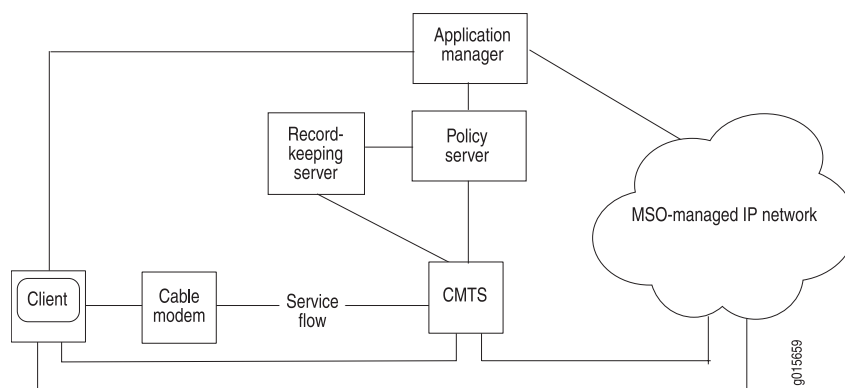
In a PCMM environment, the policy server acts as a policy decision point (PDP) and policy enforcement point (PEP) that manages the relationships between application managers and cable management termination system (CMTS) devices.

The JPS is a PCMM-compliant policy server. The JPS must be deployed in an SRC environment that satisfies these conditions:

- Organizes PCMM devices into groups (for example, one or more per POP). For redundancy, a community of two or more JPSs will manage each group of PCMM devices.
- Achieves successful state synchronization by requiring an application manager (for example, a pair of redundant SAEs) to talk to one JPS instance at a time.
- Uses IPSec connections for the network interfaces.

JPS Framework

Figure 15 on page 74 depicts the PCMM architectural framework. The JPS communicates with application managers, CMTS devices, and record-keeping servers.

Figure 15: PCMM Architectural Framework

The interactions between the various PCMM components are centered on the policy server. In the PCMM architecture, these basic interactions occur:

1. A client requests a multimedia service from an application manager.
2. Depending on the client type and its QoS signaling capabilities, the application manager relays the request to a policy server.
3. The policy server relays the request to the CMTS device and is responsible for provisioning the policies on a CMTS device.

Depending on the request, the policy server records an event for the policy request and provides that information to the record-keeping server (RKS).

4. The CMTS device performs admission control and manages network resources through Data over Cable Service Interface Specifications (DOCSIS) service flows based on the provisioned policies.
5. The RKS receives event messages from other network elements, such as the policy server or CMTS device, and acts as a short-term repository for the messages.

JPS Interfaces

The JPS has interfaces, implemented as plug-ins, to communicate with:

- Application managers, such as the SAE
- Record-keeping servers
- CMTS devices

The JPS is relatively stateless, but the individual plug-ins can be stateful.

The JPS uses the Common Open Policy Service (COPS) protocol as specified in the PacketCable Multimedia Specification PKT-SP-MM-I03-051221 for its interface connections. The JPS communicates with the CMTS device and the application manager by using a COPS over Transmission Control Protocol (TCP) connection.

Application Manager to Policy Server Interface

To allow the JPS to communicate with the application manager, this interface accepts and manages COPS over TCP connections from application managers, such as the SAE.

Policy Server to RKS Interface

To allow the JPS to communicate with a set of redundant record-keeping servers, this interface sends a policy event message to the RKS when receiving a PCMM-COPS gate control (request, delete, update) message. This interface also sends time change events to the RKS.

Policy Server to CMTS Interface

To allow the JPS to communicate with policy enforcement points (PCMM devices), this interface initiates and manages COPS over TCP connections with CMTS devices.

Chapter 10

Configuring the JPS with the SRC CLI

- Configuration Statements for the JPS on page 77
- Configuring the JPS on page 79
- Modifying the JPS Configuration on page 80
- Configuring General Properties for the JPS on page 80
- Specifying Policy Server Identifiers in Messages on page 81
- Configuring Logging Destinations for the JPS on page 82
- Configuring JPS to Store Log Messages in a File on page 82
- Configuring JPS to Send Log Messages to System Logging Facility on page 83
- Specifying Connections to the Application Managers on page 83
- Configuring Connections to RKSs on page 85
- Configuring RKS Pairs for Associated Application Managers on page 88
- Specifying Connections to CMTS Devices on page 89
- Modifying the Subscriber Configuration on page 92
- Configuring Subscriber IP Pools as IP Address Ranges on page 93
- Configuring Subscriber IP Pools as IP Subnets on page 93
- Configuring the SAE to Interact with the JPS on page 94
- Specifying Application Managers for the Policy Server on page 94
- Specifying Application Manager Identifiers for Policy Servers on page 95
- Adding Objects for Policy Servers to the Directory on page 96
- Configuring Initialization Scripts on page 97
- Enabling State Synchronization on page 97
- Using the NIC Resolver on page 98
- Managing the JPS on page 99

Configuration Statements for the JPS

Use the following configuration statements to configure the JPS at the [edit] hierarchy level.

```
slot number jps {  
    java-heap-size java-heap-size;
```

```

snmp-agent;
policy-server-id policy-server-id;
use-psid-in-gate-commands;
cmts-message-buffer-size cmts-message-buffer-size;
am-message-buffer-size am-message-buffer-size;
}
slot number jps am-interface {
    pep-id pep-id;
    listening-address listening-address;
    validate-pcmm-objects;
    message-max-length message-max-length;
    message-read-buffer-size message-read-buffer-size;
    message-write-buffer-size message-write-buffer-size;
    open-connection-timeout open-connection-timeout;
}
slot number jps cmts-interface {
    cmts-addresses [cmts-addresses...];
    keepalive-interval keepalive-interval;
    synch-despite-unreachable-pep;
    synch-despite-pre-i03-pep;
    use-ssq-ssc-with-pre-i03-pep;
    local-address local-address;
    message-max-length message-max-length;
    message-read-buffer-size message-read-buffer-size;
    message-write-buffer-size message-write-buffer-size;
    open-connection-timeout open-connection-timeout;
    connection-open-retry-interval connection-open-retry-interval;
    sent-message-timeout sent-message-timeout;
    validate-pcmm-objects;
}
slot number jps cmts-registry cmts cmts-ip ...
slot number jps cmts-registry cmts cmts-ip range-pool pool-index {
    low low;
    high high;
}
slot number jps cmts-registry cmts cmts-ip subnet-pool subnet {
    exclude [exclude];
}
slot number jps logger name ...
slot number jps logger name file {
    filter filter;
    filename filename;
    rollover-filename rollover-filename;
    maximum-file-size maximum-file-size;
}
slot number jps logger name syslog {
    filter filter;
    host host;
    facility facility;
    format format;
}
slot number jps rks-interface {
    element-id element-id;
    local-address local-address;
    local-port local-port;

```

```

retry-interval retry-interval;
local-timeout local-timeout;
mso-data mso-data;
mso-domain-name mso-domain-name;
default-rks-pair default-rks-pair;
pending-rks-event-max-size pending-rks-event-max-size;
pending-rks-event-max-age pending-rks-event-max-age;
held-decs-max-size held-decs-max-size;
held-decs-max-age held-decs-max-age;
bcid-cache-size bcid-cache-size;
bcid-cache-age bcid-cache-age;
use-default-when-am-requests-unconfigured-rks;
}
slot number jps rks-interface am am-name {
    am-id am-id;
    rks-pair-name rks-pair-name;
    trusted;
}
slot number jps rks-interface rks-pair rks-pair-name {
    primary-address primary-address;
    primary-port primary-port;
    secondary-address secondary-address;
    secondary-port secondary-port;
}

```

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

Configuring the JPS

You can modify the JPS configuration, which includes configuring the logging destinations and connections to the JPS interfaces. Any configuration changes will be applied within 15 seconds.

Before you configure the JPS, deploy an SRC-managed PCMM network. For more information about PCMM and the SRC software, see *Overview of a PCMM Environment*.

You can configure the subscriber configuration, which maps a subscriber address to the CMTS address.

The tasks to configure the JPS for a cable network environment are:

- Modifying the JPS Configuration
- Modifying the Subscriber Configuration

In addition to configuring the JPS, you might need to perform these tasks:

- Configuring the SAE to Interact with the JPS
- Using the NIC Resolver

Modifying the JPS Configuration

To modify the current JPS configuration:

1. Configure general properties for the JPS, including Java heap memory, maximum number of buffered messages for CMTS and application manager destinations, and policy server identifiers.

See [Configuring General Properties for the JPS](#) .

See [Specifying Policy Server Identifiers in Messages](#) .

2. Configure logging destinations for the JPS.

See [Configuring Logging Destinations for the JPS](#) .

3. Configure the connections to the JPS interfaces.

See [Specifying Connections to the Application Managers](#) .

See [Specifying Connections to CMTS Devices](#) .

Configuring General Properties for the JPS

Use the following configuration statements to configure general properties for the JPS:

```
slot number jps {
  java-heap-size java-heap-size;
  snmp-agent;
  cmts-message-buffer-size cmts-message-buffer-size;
  am-message-buffer-size am-message-buffer-size;
}
```

To configure general properties for the JPS:

1. From configuration mode, access the configuration statement that configures the general properties.

```
user@host# edit slot 0 jps
```

2. (Optional) Specify the maximum amount of memory available to the Java Runtime Environment (JRE).

```
[edit slot 0 jps]
user@host# set java-heap-size java-heap-size
```

3. (Optional) Enable the JPS to communicate with the SNMP agent.

```
[edit slot 0 jps]
user@host# set snmp-agent
```

4. (Optional) Specify the maximum number of messages buffered for each CMTS destination.

```
[edit slot 0 jps]
user@host# set cmts-message-buffer-size cmts-message-buffer-size
```

5. (Optional) Specify the maximum number of messages buffered for each application manager destination.

```
[edit slot 0 jps]
user@host# set am-message-buffer-size am-message-buffer-size
```

6. (Optional) Verify your configuration.

```
[edit slot 0 jps]
user@host# show
```

Specifying Policy Server Identifiers in Messages

Use the following configuration statements to configure policy server identifiers for the JPS:

```
slot number jps {
  policy-server-id policy-server-id;
  use-psid-in-gate-commands;
}
```

To configure policy server identifiers for the JPS:

1. From configuration mode, access the configuration statement that configures the policy server identifiers.

```
user@host# edit slot 0 jps
```

2. (Optional) Specify the policy server identifier so that the JPS can be identified in messages sent to CMTS devices.

```
[edit slot 0 jps]
user@host# set policy-server-id policy-server-id
```

3. (Optional) Configure the JPS so that the policy server identifier is specified in messages sent to the RKS.

```
[edit slot 0 jps]
user@host# set use-psid-in-gate-commands
```

When the JPS is communicating only with PCMM I03 CMTS devices, the value must be true. When the JPS is communicating with any pre-PCMM I03 CMTS devices, the value must be false.

4. (Optional) Verify your configuration.

```
[edit slot 0 jps]
user@host# show
```

Configuring Logging Destinations for the JPS

By default, the JPS has four logging destinations.

Use the following configuration statements to configure logging destinations for the JPS:

```
slot number jps logger name ...
slot number jps logger name file {
    filter filter;
    filename filename;
    rollover-filename rollover-filename;
    maximum-file-size maximum-file-size;
}
slot number jps logger name syslog {
    filter filter;
    host host;
    facility facility;
    format format;
}
```

Configuring JPS to Store Log 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 log2 is configured.

```
user@host# edit slot 0 jps logger log2 file
```

2. Specify the properties for the logging destination.

```
[edit slot 0 jps logger log2 file]
user@host# set ?
```

For more information about configuring properties for the logging destination, see Overview of Logging for SRC Components.

3. (Optional) Verify your configuration.

```
[edit slot 0 jps logger log2]
user@host# show
file {
    filter !NoAckRksEvent,/info-;
    filename var/log/jps_info.log;
    rollover-filename var/log/jps_info.alt;
    maximum-file-size 2000000000;
}
```

Configuring JPS to Send Log 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 log5 is configured.

```
user@host# edit slot 0 jps logger log5 syslog
```

2. Specify the properties for the logging destination.

```
[edit slot 0 jps logger log5 syslog]
user@host# set ?
```

For more information about configuring properties for the logging destination, see Overview of Logging for SRC Components.

3. (Optional) Verify your configuration.

```
[edit slot 0 jps logger log5]
user@host# show
```

Specifying Connections to the Application Managers

Use the following configuration statement to configure the application manager-to-policy server interface (PKT-MM3) so that the policy server can communicate with application managers:

```
slot number jps am-interface {
  pep-id pep-id;
  listening-address listening-address;
  validate-pcmm-objects;
  message-max-length message-max-length;
  message-read-buffer-size message-read-buffer-size;
  message-write-buffer-size message-write-buffer-size;
  open-connection-timeout open-connection-timeout;
}
```

To configure the connections to the application managers:

1. From configuration mode, access the configuration statement that configures the application manager-to-policy server interface.

```
user@host# edit slot 0 jps am-interface
```

2. (Optional) Specify the network-wide unique identifier for this JPS instance.

```
[edit slot 0 jps am-interface]
user@host# set pep-id pep-id
```

Changes apply only to COPS connections that are established after you make the change.

3. (Optional) Specify the local IP address on which the JPS listens for incoming connections from application managers.

```
[edit slot 0 jps am-interface]
user@host# set listening-address listening-address
```

Changes take effect only after you restart the JPS (see Restarting the JPS).

4. (Optional) Specify whether to validate PCMM objects received from PDPs.

```
[edit slot 0 jps am-interface]
user@host# set validate-pcmm-objects
```

5. (Optional) Specify the maximum length of incoming messages.

```
[edit slot 0 jps am-interface]
user@host# set message-max-length message-max-length
```

6. (Optional) Specify the size of message read buffer.

```
[edit slot 0 jps am-interface]
user@host# set message-read-buffer-size message-read-buffer-size
```

7. (Optional) Specify the size of message write buffer.

```
[edit slot 0 jps am-interface]
user@host# set message-write-buffer-size message-write-buffer-size
```

8. (Optional) Specify the maximum time to wait for the initial PCMM messages to be exchanged after a TCP connection is established.

```
[edit slot 0 jps am-interface]
user@host# set open-connection-timeout open-connection-timeout
```

The connection is dropped when initial PCMM messages are not exchanged within this time period.

9. (Optional) Verify your configuration.

```
[edit slot 0 jps am-interface]
user@host# show
pep-id SDX-JPS;
listening-address ;
validate-pcmm-objects;
message-max-length 204800;
message-read-buffer-size 1000000;
message-write-buffer-size 1000000;
open-connection-timeout 5;
```

Configuring Connections to RKSs

To configure connections to RKSs:

1. Specifying Connections to RKSs on page 85
2. Configuring RKS Pairs on page 87

Specifying Connections to RKSs

To configure the policy server-to-RKS interface (PKT-MM4) so that policy events can be sent to the RKS, you can configure RKS pairs (see “Configuring RKS Pairs” on page 87) and their associated application managers (see “Configuring RKS Pairs for Associated Application Managers” on page 88).

Use the following configuration statement to configure the policy server-to-RKS interface:

```
slot number jps rks-interface {
  element-id element-id;
  local-address local-address;
  local-port local-port;
  retry-interval retry-interval;
  local-timeout local-timeout;
  mso-data mso-data;
  mso-domain-name mso-domain-name;
  default-rks-pair default-rks-pair;
  pending-rks-event-max-size pending-rks-event-max-size;
  pending-rks-event-max-age pending-rks-event-max-age;
  held-decs-max-size held-decs-max-size;
  held-decs-max-age held-decs-max-age;
  bcid-cache-size bcid-cache-size;
  bcid-cache-age bcid-cache-age;
  use-default-when-am-requests-unconfigured-rks;
}
```

To configure the policy server-to-RKS interface:

1. From configuration mode, access the configuration statement that configures the policy server-to-RKS interface.

```
user@host# edit slot 0 jps rks-interface
```

2. Enter for RKS event origin.

```
[edit slot 0 jps rks-interface]
user@host# set element-id element-id
```

3. (Optional) Specify the source IP address that the plug-in uses to communicate with the RKS.

```
[edit slot 0 jps rks-interface]
user@host# set local-address local-address
```

If no value is specified and there is more than one local address, the JPS randomly selects a local address to be used as the source address.

4. (Optional) Specify the source UDP port or a pool of ports that the plug-in uses to communicate with the RKS.

```
[edit slot 0 jps rks-interface]
user@host# set local-port local-port
```

5. (Optional) Specify the time the JPS waits for a response from an RKS before it resends the packet.

```
[edit slot 0 jps rks-interface]
user@host# set retry-interval retry-interval
```

The JPS keeps sending packets until either the RKS acknowledges the packet or the maximum timeout is reached.

6. (Optional) Specify the maximum time the JPS waits for a response from an RKS.

```
[edit slot 0 jps rks-interface]
user@host# set local-timeout local-timeout
```

7. (Optional) Specify the MSO-defined data in the financial entity ID (FEID) attribute, which is included in event messages.

```
[edit slot 0 jps rks-interface]
user@host# set mso-data mso-data
```

8. (Optional) Specify the MSO domain name in the FEID attribute that uniquely identifies the MSO for billing and settlement purposes.

```
[edit slot 0 jps rks-interface]
user@host# set mso-domain-name mso-domain-name
```

9. (Optional) Specify the default RKS pair that the JPS uses unless an RKS pair is configured for a given application manager.

```
[edit slot 0 jps rks-interface]
user@host# set default-rks-pair default-rks-pair
```

10. (Optional) Specify the maximum number of RKS events waiting for Gate-Set-Ack, Gate-Set-Err, Gate-Del-Ack, and Gate-Del-Err messages.

```
[edit slot 0 jps rks-interface]
user@host# set pending-rks-event-max-size pending-rks-event-max-size
```

11. (Optional) Specify the oldest age of RKS events waiting for Gate-Set-Ack, Gate-Set-Err, Gate-Del-Ack, and Gate-Del-Err messages.

```
[edit slot 0 jps rks-interface]
user@host# set pending-rks-event-max-age pending-rks-event-max-age
```

The maximum age must be greater than sent-message-timeout of the corresponding CMTS interface.

12. (Optional) Specify the maximum number of outstanding Gate-Info requests.

```
[edit slot 0 jps rks-interface]
user@host# set held-decs-max-size held-decs-max-size
```

13. (Optional) Specify the oldest age of outstanding Gate-Info requests.

```
[edit slot 0 jps rks-interface]
user@host# set held-decs-max-age held-decs-max-age
```

The maximum age must be greater than sent-message-timeout of the corresponding CMTS interface.

14. (Optional) Specify the size of billing correlation ID (BCID) cache.

```
[edit slot 0 jps rks-interface]
user@host# set bcid-cache-size bcid-cache-size
```

15. (Optional) Specify the oldest age of billing correlation ID (BCID) in cache.

```
[edit slot 0 jps rks-interface]
user@host# set bcid-cache-age bcid-cache-age
```

16. (Optional) Specify whether the default RKS pair is used when an application manager requests the use of an unconfigured RKS pair.

```
[edit slot 0 jps rks-interface]
user@host# set use-default-when-am-requests-unconfigured-rks
```

17. (Optional) Verify your configuration.

```
[edit slot 0 jps rks-interface]
user@host# show
```

Configuring RKS Pairs

By default, the JPS has four RKS pairs. All parameters that share the same RKS pair name configure the connection to that RKS pair. Any configured RKS pair can be used as the value for the default RKS pair or the RKS pair associated with a specific application manager.



NOTE: When running more than one JPS in a group to provide redundancy, all the JPSs in that group must have same RKS pair configuration (including the default RKS pair and any configured RKS pairs associated with a specific application manager).

Use the following configuration statement to configure the RKS pair:

```
slot number jps rks-interface rks-pair rks-pair-name {
```

```

primary-address primary-address;
primary-port primary-port;
secondary-address secondary-address;
secondary-port secondary-port;
}

```

To configure the RKS pair:

1. From configuration mode, access the configuration statement that configures the RKS pair. In this sample procedure, the RKS pair called pair1 is configured.

```

user@host# edit slot 0 jps rks-interface rks-pair pair1

```

2. Specify the IP address of the primary RKS for this RKS pair.

```

[edit slot 0 jps rks-interface rks-pair pair1]
user@host# set primary-address primary-address

```

If no value is specified, the RKS pair is not defined.

3. (Optional) Specify the UDP port on the primary RKS to which the JPS sends events.

```

[edit slot 0 jps rks-interface rks-pair pair1]
user@host# set primary-port primary-port

```

4. (Optional) Specify the IP address of the secondary RKS for this RKS pair.

```

[edit slot 0 jps rks-interface rks-pair pair1]
user@host# set secondary-address secondary-address

```

5. (Optional) Specify the UDP port on the secondary RKS to which the JPS sends events.

```

[edit slot 0 jps rks-interface rks-pair pair1]
user@host# set secondary-port secondary-port

```

6. (Optional) Verify your configuration.

```

[edit slot 0 jps rks-interface rks-pair pair1]
user@host# show
primary-address ;
primary-port 1813;
secondary-address ;
secondary-port 1813;

```

Configuring RKS Pairs for Associated Application Managers

By default, the JPS has four associated application managers. All parameters that share the same application manager name configure the RKS pair to which events associated with a specific application manager are sent.

Use the following configuration statement to configure the associated application manager:

```
slot number jps rks-interface am am-name {
    am-id am-id;
    rks-pair-name rks-pair-name;
    trusted;
}
```

To configure the associated application manager:

1. From configuration mode, access the configuration statement that configures the RKS pair for the associated application manager. In this sample procedure, the application manager name called 1 is configured.

```
user@host# edit slot 0 jps rks-interface am 1
```

2. Specify the identifier of the application manager.

```
[edit slot 0 jps rks-interface am 1]
user@host# set am-id am-id
```

If no value is specified, the RKS pair configuration is not defined for this application manager. If you must set `trusted` to true without defining the RKS pair configuration, you must specify a value for `am-id` and not specify a value for `rks-pair-name`.

3. (Optional) Specify the RKS pair that the JPS will send events to when those events are triggered by gate transitions associated with the application manager specified by `am-id` with the same application manager name (`am-name`).

```
[edit slot 0 jps rks-interface am 1]
user@host# set rks-pair rks-pair-name
```

If no value is specified, the RKS pair configuration is not defined for this application manager. Use when you must set `trusted` to true without defining the RKS pair configuration.

4. (Optional) Specify whether this application manager is a trusted network element to the JPS.

```
[edit slot 0 jps rks-interface am 1]
user@host# set trusted
```

5. (Optional) Verify your configuration.

```
[edit slot 0 jps rks-interface am 1]
user@host# show
```

Specifying Connections to CMTS Devices

Use the following configuration statement to configure the policy server-to-CMTS interface (PKT-MM2) so that the policy server can communicate with CMTS devices:

```

slot number jps cmts-interface {
    cmts-addresses [cmts-addresses...];
    keepalive-interval keepalive-interval;
    synch-despite-unreachable-pep;
    synch-despite-pre-i03-pep;
    use-ssq-ssc-with-pre-i03-pep;
    local-address local-address;
    message-max-length message-max-length;
    message-read-buffer-size message-read-buffer-size;
    message-write-buffer-size message-write-buffer-size;
    open-connection-timeout open-connection-timeout;
    connection-open-retry-interval connection-open-retry-interval;
    sent-message-timeout sent-message-timeout;
    validate-pcmm-objects;
}

```

To configure the policy server-to-CMTS interface:

1. From configuration mode, access the configuration statement that configures the policy server-to-CMTS interface.

```
user@host# edit slot 0 jps cmts-interface
```

2. Specify the IP addresses of all the CMTS devices to which the JPS will try to connect.

```

[edit slot 0 jps cmts-interface]
user@host# set cmts-addresses [cmts-addresses...]

```

3. (Optional) Specify the interval between keepalive messages sent from the COPS client (CMTS device) to the COPS server (the JPS). Changes apply only to COPS connections that are established after you make the change.

```

[edit slot 0 jps cmts-interface]
user@host# set keepalive-interval keepalive-interval

```

A value of 0 means that no keepalive messages will be exchanged between the CMTS device and the JPS.

4. (Optional) Specify whether synchronization proceeds when the JPS receives a synchronization request from an application manager (such as the SAE) and the JPS is not connected to a CMTS device to which it should be connected.

```

[edit slot 0 jps cmts-interface]
user@host# set synch-despite-unreachable-pep

```

5. (Optional) Specify whether synchronization proceeds when the JPS receives a synchronization request from an application manager (such as the SAE) and the JPS is connected to a pre-PCMM I03 CMTS device.

```

[edit slot 0 jps cmts-interface]
user@host# set synch-despite-pre-i03-pep

```

6. (Optional) Specify whether synchronization includes both pre-PCMM I03 and PCMM I03 CMTS devices when the JPS receives a synchronization request from an application manager (such as the SAE) and the JPS is connected to a pre-PCMM I03 CMTS device. Relevant only when at least one pre-PCMM I03 CMTS device is connected and `sync-despite-pre-i03-pep` is specified as true.

```
[edit slot 0 jps cmts-interface]
user@host# set use-ssq-ssc-with-pre-i03-pep
```

7. (Optional) Specify the source IP address that the JPS uses to communicate with CMTS devices.

```
[edit slot 0 jps cmts-interface]
user@host# set local-address local-address
```

If no value is specified and there is more than one local address, a random local address is used as the source address.

8. (Optional) Specify the maximum length of incoming messages.

```
[edit slot 0 jps cmts-interface]
user@host# set message-max-length message-max-length
```

9. (Optional) Specify the size of message read buffer.

```
[edit slot 0 jps cmts-interface]
user@host# set message-read-buffer-size message-read-buffer-size
```

10. (Optional) Specify the size of message write buffer.

```
[edit slot 0 jps cmts-interface]
user@host# set message-write-buffer-size message-write-buffer-size
```

11. (Optional) Specify the maximum time to wait for the initial PCMM messages to be exchanged after a TCP connection is established.

```
[edit slot 0 jps cmts-interface]
user@host# set open-connection-timeout open-connection-timeout
```

The connection is dropped when initial PCMM messages are not exchanged within this time period.

12. (Optional) Specify the time to wait before the JPS tries to reconnect to CMTS devices.

```
[edit slot 0 jps cmts-interface]
user@host# set connection-open-retry-interval connection-open-retry-interval
```

13. (Optional) Specify the maximum time to wait for the sent messages to be exchanged after a TCP connection is established.

```
[edit slot 0 jps cmts-interface]
user@host# set sent-message-timeout sent-message-timeout
```

This value must be less than the held-decs-max-age and pending-rks-event-max-age values for the corresponding RKS interface.

14. (Optional) Specify whether to validate PCMM objects received from PDPs.

```
[edit slot 0 jps cmts-interface]
user@host# set validate-pcmm-objects
```

15. (Optional) Verify your configuration.

```
[edit slot 0 jps cmts-interface]
user@host# show
cmts-addresses ;
keepalive-interval 60;
synch-despite-unreachable-pep;
synch-despite-pre-i03-pep;
local-address ;
message-max-length 204800;
message-read-buffer-size 1000000;
message-write-buffer-size 1000000;
open-connection-timeout 5;
connection-open-retry-interval 60;
sent-message-timeout 60;
validate-pcmm-objects;
```

Modifying the Subscriber Configuration

To locate the CMTS device associated with a subscriber, the JPS maps the subscriber IP address in a message to the CMTS IP address to which the message must be delivered. This mapping specifies the subscriber IP pools associated with CMTS devices.

Use the following configuration statements to configure a CMTS device to which the JPS can connect and the pools of subscriber IP addresses that are managed by the CMTS device:

```
slot number jps cmts-registry cmts cmts-ip ...
slot number jps cmts-registry cmts cmts-ip range-pool pool-index {
    low low;
    high high;
}
slot number jps cmts-registry cmts cmts-ip subnet-pool subnet {
    exclude [exclude];
}
```

Tasks to modify subscriber configuration are:

1. Configuring Subscriber IP Pools as IP Address Ranges
2. Configuring Subscriber IP Pools as IP Subnets

Configuring Subscriber IP Pools as IP Address Ranges

To configure subscriber IP pools that are managed by the CMTS device as IP address ranges:

1. From configuration mode, access the configuration statement that configures the CMTS device to which the JPS can connect.

```
user@host# edit slot 0 jps cmts-registry cmts cmts-ip range-pool pool-index
```

Specify the IP address of the CMTS device and the address range pool index.

2. Specify the first IP address in the IP range for the pool of subscriber IP addresses that are managed by the CMTS device.

```
[edit slot 0 jps cmts-registry cmts cmts-ip range-pool pool-index]
user@host# set low low
```

3. Specify the last IP address in the IP range for the pool of subscriber IP addresses that are managed by the CMTS device.

```
[edit slot 0 jps cmts-registry cmts cmts-ip range-pool pool-index]
user@host# set high high
```

4. (Optional) Verify your configuration.

```
[edit slot 0 jps cmts-registry]
user@host# show
```

Configuring Subscriber IP Pools as IP Subnets

To configure subscriber IP pools that are managed by the CMTS device as IP subnets:

1. From configuration mode, access the configuration statement that configures the CMTS device to which the JPS can connect.

```
user@host# edit slot 0 jps cmts-registry cmts cmts-ip subnet-pool subnet
```

Specify the IP address of the CMTS device and the IP address and mask of the subnet for the pool of subscriber IP addresses.

2. (Optional) Specify the IP addresses of the subnet that are excluded from the subscriber IP pool managed by the CMTS device.

```
[edit slot 0 jps cmts-registry cmts cmts-ip subnet-pool subnet]
user@host# set exclude [exclude...]
```

3. (Optional) Verify your configuration.

```
[edit slot 0 jps cmts-registry]
user@host# show
```

Configuring the SAE to Interact with the JPS

You must configure the SAE as an application manager to allow it to interact with PCMM-compliant policy servers. The policy server acts as a policy decision point that manages the relationships between application managers and CMTS devices. Policy servers that manage the same group of CMTS devices are grouped together and are simultaneously active. The policy server group provides a way for the SAE to communicate with any CMTS device that is managed by a policy server in the policy server group. To provide redundancy, the SAEs are grouped in an SAE community that connects to a policy server group. Only one of the SAEs in the SAE community is active. The active SAE establishes connections to all the policy servers in the policy server group. The active SAE will fail over to a redundant SAE only when it loses the connection to all the policy servers in the policy server group. State synchronization enables the SAE to synchronize its state with all the CMTS devices connected to a policy server group.

The tasks to configure the SAE as an application manager are:

- Specifying Application Managers for the Policy Server
- Specifying Application Manager Identifiers for Policy Servers
- Adding Objects for Policy Servers to the Directory
- Configuring Initialization Scripts
- Enabling State Synchronization

Specifying Application Managers for the Policy Server

To specify the SAE community that connects to a policy server group, you need to add an application manager group object to the directory.

Use the following configuration statements to specify the application manager for the policy server:

```
shared network application-manager-group name {
  description description;
  application-manager-id application-manager-id;
  connected-sae [connected-sae...];
  pdp-group pdp-group;
  local-address-pools [local-address-pools...];
  managing-sae-ior managing-sae-ior;
}
```

To add an application manager group:

1. From configuration mode, access the configuration statement that specifies the application managers.

```
user@host# edit shared network application-manager-group name
```

2. (Optional) Specify information about the SAE community.

```
[edit shared network application-manager-group name]
```

user@host# **set description** *description*

3. (Optional) Specify the unique identifier within the domain of the service provider for the application manager that handles the service session (Application Manager Tag) as a 2-byte unsigned integer.

[edit shared network application-manager-group *name*]
user@host# **set application-manager-id** *application-manager-id*

4. (Optional) Specify the SAEs that are connected to the specified policy server group. This list becomes the community of SAEs.

[edit shared network application-manager-group *name*]
user@host# **set connected-sae** [*connected-sae...*]

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

5. (Optional) Specify the name of the policy server group associated with this SAE community.

[edit shared network application-manager-group *name*]
user@host# **set pdp-group** *pdp-group*

6. (Optional) Specify the list of IP address pools that the specified PDP group currently manages and stores.

[edit shared network application-manager-group *name*]
user@host# **set local-address-pools** *local-address-pools*

You must configure a local address pool if you are using the NIC so that the NIC can resolve the IP-to-SAE mapping. See Using the NIC Resolver.

7. (Optional) Specify the Common Object Request Broker Architecture (CORBA) reference for the SAE managing this policy server group.

[edit shared network application-manager-group *name*]
user@host# **set managing-sae-ior** *managing-sae-ior*

The **amlorPublisher** script provides this information when the SAE connects to the policy server. If you do not select this script when configuring initialization scripts, enter a value. For information about configuring initialization scripts, see Configuring Initialization Scripts.

Specifying Application Manager Identifiers for Policy Servers

The application manager identifier (AMID) identifies the application manager (such as the SAE) in messages sent to and from the policy server. The SAE constructs the

AMID value by concatenating two fields: Application Manager Tag and Application Type.

The Application Manager Tag value is obtained from the specification of application managers for policy servers. See *Specifying Application Managers for the Policy Server*.

The Application Type value is obtained during service activation from the specification of the PCMM Application Type value when you configure normal services.

For more information about configuring services, see *Adding a Normal Service (SRC CLI)*.

Adding Objects for Policy Servers to the Directory

To communicate with policy servers, the SAE creates and manages pseudointerfaces that it associates with a policy decision point object in the directory. Each policy server in the SRC network must appear in the directory as a policy decision point object.

Use the following configuration statements to specify the policy server as a policy decision point:

```
shared network policy-decision-point name {
  description description;
  pdp-address pdp-address;
  pdp-group pdp-group;
}
```

To add a policy server to the directory with the SRC CLI:

1. From configuration mode, access the configuration statement that configures the policy decision point.

```
user@host# edit shared network policy-decision-point name
```

2. (Optional) Specify information about the policy server.

```
[edit shared network policy-decision-point name]
user@host# set description description
```

3. (Optional) Specify the IP address of the policy server. The SAE uses this address to establish a COPS connection with the policy server.

```
[edit shared network policy-decision-point name]
user@host# set pdp-address pdp-address
```

4. (Optional) Specify the name of the policy server group.

```
[edit shared network policy-decision-point name]
user@host# set pdp-group pdp-group
```

5. Create an SAE community for the policy servers. See Specifying Application Managers for the Policy Server .

Configuring Initialization Scripts

When the SAE establishes a connection with a policy server, it runs an initialization script to customize the setup of the connection.

Use the following configuration statement to configure the initialization script:

```
shared sae configuration driver scripts {
    pcmm pcmm;
}
```

To configure initialization scripts for the SAE:

1. From configuration mode, access the configuration statement that configures the initialization scripts.

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

2. Specify the initialization script for a PCMM environment.

```
[edit shared sae configuration driver scripts]
user@host# set pcmm pcmm
```

The script is run when the connection between a policy server and the SAE is established and again when the connection is dropped. For the JPS, we recommend setting this value to `amIorPublisher`.

Enabling State Synchronization

State synchronization is achieved when the SAE is required to communicate with the policy server over the COPS connection.

Use the following configuration statement to configure state synchronization:

```
shared sae configuration driver pcmm {
    disable-full-sync;
    disable-pcmm-iO3-policy;
    session-recovery-retry-interval session-recovery-retry-interval;
}
```

To enable state synchronization with policy servers:

1. From configuration mode, access the configuration statement that configures the PCMM device driver.

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

2. Specify whether state synchronization with the PCMM policy servers is disabled.

```
[edit shared sae configuration driver pcmm]
user@host# set disable-full-sync
```

When using other PCMM-compliant policy servers (instead of the JPS), we recommend setting this value to true.

3. Specify whether PCMM I03 policies are disabled when the SAE is deployed with pre-PCMM I03 CMTS devices.

```
[edit shared sae configuration driver pcmm]
user@host# set disable-pcmm-i03-policy
```

When there are pre-PCMM I03 CMTS devices in the network, you must set this value to true.

4. Specify the time interval between attempts by the SAE to restore service sessions that are still being recovered in the background when state synchronization completes with a state-data-incomplete error.

```
[edit shared sae configuration driver pcmm]
user@host# set session-recovery-retry-interval session-recovery-retry-interval
```

We recommend setting this value to 3600000 (1 hour) or longer.

Using the NIC Resolver

If you are using the NIC to map the subscriber IP address to the SAE, you need to configure a NIC host. The NIC system uses IP address pools to map IP addresses to SAEs. You configure the local address pools in the application manager configuration for a policy server group. These pools are published in the NIC. The NIC maps subscriber IP addresses in requests received through the portal or Advanced Services Gateway to the policy server group that currently manages that CMTS device. For information about configuring the SAE for policy servers, see *Specifying Application Managers for the Policy Server*.

The OnePopPcmm sample configuration data supports this scenario for a 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 OnePopPcmm configuration supports one point of presence (POP). NIC replication can be used to provide high availability. The realm for this configuration accommodates the situation in which IP pools are configured locally on each application manager group 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 following agents collect information for resolvers in this realm:

- Directory agent PoolVr collects and publishes information about the mappings of IP pools to the policy server group.
- Directory agent VrSaeld collects and publishes information about the mappings of policy server groups to SAEs.

For more information about configuring the NIC, see [Configuring the NIC \(SRC CLI\)](#).

Managing the JPS

After you have installed the JPS and applied the local configuration of the JPS, you can perform these tasks:

- Starting the JPS on page 99
- Restarting the JPS on page 99
- Stopping the JPS on page 99
- Displaying JPS Status on page 99

Starting the JPS

You must start the JPS when you install the JPS without rebooting the JPS host.

To start the JPS:

```
user@host> enable component jps
```

The system responds with a start message. If the JPS is already running, the system responds with a warning message.

Restarting the JPS

To restart the JPS:

```
user@host> restart component jps
```

The system responds with a start message. If the JPS is already running, the system responds with a shutdown message and then a start message.

Stopping the JPS

To stop the JPS:

```
user@host> disable component jps
```

The system responds with a shutdown message. If the JPS is not running when you issue the command, the system responds with the command prompt.

To start the JPS, see [Starting the JPS](#).

Displaying JPS Status

Purpose Display the JPS status.

Action user@host> show component

The system responds with a status message.

Chapter 11

Monitoring the JPS with the SRC CLI

- Monitoring the JPS on page 101
- Viewing Server Process Information on page 101
- Viewing JPS State on page 102

Monitoring the JPS

Purpose Monitor the following JPS information:

- The basic health indicators for the server process
- The current state of the JPS, such as the current network connections or recent performance statistics

Action user@host> **show jps statistics**

- Related Topics**
- Viewing Server Process Information
 - Viewing JPS State
 - Overview of the JPS

Viewing Server Process Information

Purpose View information about the server process.

Action user@host> **show jps statistics process**

- Related Topics**
- Monitoring the JPS
 - Viewing JPS State
 - Overview of the JPS

Viewing JPS State

You can monitor the current state of the JPS by:

1. Viewing Performance Statistics for the JPS Interfaces on page 102
2. Viewing Network Connections for the Application Manager on page 102
3. Viewing Network Connections for the CMTS Device on page 102
4. Viewing Performance Statistics for the CMTS Locator on page 103
5. Viewing Message Handler Information on page 103

Viewing Performance Statistics for the JPS Interfaces

Purpose View performance statistics for JPS interfaces.

Action To view recent performance statistics for the application manager-to-policy server interface:

```
user@host> show jps statistics am
```

To view recent performance statistics for the policy server-to-CMTS interface:

```
user@host> show jps statistics cmts
```

To view recent performance statistics for the policy server-to-RKS interface:

```
user@host> show jps statistics rks
```

Viewing Network Connections for the Application Manager

Purpose View network connections for the application manager.

Action To view information about the current JPS network connections for all the application managers:

```
user@host> show jps statistics am connections
```

To view information about the current JPS network connections for a specific application manager:

```
user@host> show jps statistics am connections ip-address ip-address
```

Enter all or part of the IP address to list connections for all matching addresses.

Viewing Network Connections for the CMTS Device

Purpose View network connections for the CMTS Device.

Action To view information about the current JPS connections for all the CMTS devices:

```
user@host> show jps statistics cmts connections
```

To view information about the current JPS connections for a specific CMTS device:

```
user@host> show jps statistics cmts connections ip-address ip-address
```

Enter all or part of the IP address to list connections for all matching addresses.

Viewing Performance Statistics for the CMTS Locator

Purpose View information about the recent performance statistics for the CMTS locator.

Action user@host> **show jps statistics cmts-locator**

Viewing Message Handler Information

Purpose View message handler information.

Action To view information about the JPS message handler and message flows:

```
user@host> show jps statistics message-handler
user@host> show jps statistics message-handler message-flow
```

To view information about specific JPS message flows:

```
user@host> show jps statistics message-handler message-flow id id
```

Enter all or part of the message flow identifier to list all matching message flows.

Chapter 12

Monitoring the JPS with the C-Web Interface

- Viewing Information About the JPS Server Process with the C-Web Interface on page 105
- Viewing JPS AM Statistics with the C-Web Interface on page 106
- Viewing JPS AM Connections with the C-Web Interface on page 107
- Viewing JPS CMTS Statistics with the C-Web Interface on page 107
- Viewing JPS CMTS Connections with the C-Web Interface on page 108
- Viewing JPS CMTS Locator Statistics with the C-Web Interface on page 109
- Viewing JPS Message Handler Statistics with the C-Web Interface on page 109
- Viewing JPS Message Flow Statistics with the C-Web Interface on page 110
- Viewing JPS RKS Statistics with the C-Web Interface on page 111

Viewing Information About the JPS Server Process with the C-Web Interface

Purpose View information about the JPS server process.

Action Click **JPS > Statistics > Process**.

The Statistics/Process pane displays the JPS server process information.

The screenshot shows the JPS Monitor interface. The top navigation bar includes 'Monitor', 'Configure', 'Diagnose', and 'Manage'. The user is logged in as 'admin'. The left sidebar lists various components: CLI, Component, Date, Disk, Interfaces..., JPS, NIC, NTP, Redirect Server, Route..., SAE, Security, and System. The main content area displays 'JPS Statistics / Process' with a table of server process statistics.

JPS Server Process	
JPS server up time(seconds)	1250
JPS server up since	Tue Aug 07 12:13:03 EDT 2007
JPS server thread(s)	33
Heap used(byte)	10547088 (3%)
Heap limit(byte)	400000000

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

Related Topics ■ Overview of the JPS

Viewing JPS AM Statistics with the C-Web Interface

Purpose View information about recent performance statistics for the application manager-to-policy server interface.

Action Click **JPS > Statistics > AM**.

The Statistics/AM pane displays performance statistics for the application manager-to-policy server interface.

The screenshot shows the JPS Monitor interface with the 'JPS' component selected. The main content area displays 'JPS AM Interface (PKT-MM-3)' with a table of connection statistics.

JPS AM Interface (PKT-MM-3)	
Connections opened	0
Connections closed	0

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

Related Topics ■ Overview of the JPS

Viewing JPS AM Connections with the C-Web Interface

Purpose View information about the current JPS network connections for the application manager.

Action 1. Click **JPS > Statistics > AM > Connections**.

The Statistics/AM/Connections pane appears.

2. In the IP Address box, enter the IP address, or leave the box blank to display all AM connections.
3. Click **OK**.

The Statistics/AM/Connections pane displays the AM connection statistics.

Related Topics ■ Overview of the JPS

Viewing JPS CMTS Statistics with the C-Web Interface

Purpose View information about recent performance statistics for the policy server-to-CMTS interface.

Action Click **JPS > Statistics > CMTS**.

The Statistics/CMTS pane displays statistics for the policy server-to-CMTS interface.

The screenshot shows the JPS CMTS Interface (PKT-MM-2) with the following statistics:

Component	Value
Connections opened	0
Connections closed	0
Sync-Request/SSQ broadcasts	0
Avg sync time (last 10 syncs, ms)	0
Timed out syncs	0

The interface includes a navigation menu on the left with options like CLI, Component, Date, Disk, Interfaces..., JPS, NIC, NTP, Redirect Server, Route..., SAE, Security, and System. The top bar shows 'Logged in as: admin' and buttons for Refresh, Preferences, About, and Logout. The footer contains copyright information for Juniper Networks, Inc. and the Juniper logo.

Related Topics ■ Overview of the JPS

Viewing JPS CMTS Connections with the C-Web Interface

Purpose View information about the current JPS network connections for the CMTS device.

Action 1. Click **JPS > Statistics > CMTS > Connections**.

The Statistics/CMTS/Connections pane appears.

The screenshot shows the JPS CMTS Connections pane with the following elements:

- Navigation Menu:** ACP, CLI, Component, Date, Disk, Interfaces..., Iptables..., JPS, NIC, NTP, Redirect Server, Route..., SAE, Security, System.
- Page Header:** Monitor, Configure, Diagnose, Manage. Logged in as: admin. Refresh, Preferences, About, Logout.
- Page Title:** JPS Statistics / CMTS / Connections
- Form:**
 - Ip Address:** A text box for filtering connections by IP address.
 - Buttons:** OK, Reset.
 - Help Text:** IP address for the CMTS device. Value: All or part of the IP address. If the IP address filter is not specified, all CMTS devices are selected. Default: No value.
- Footer:** Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy. Juniper Your Net.

2. In the IP Address box, enter the IP address, or leave the box blank to display all CMTS connections.
3. Click **OK**.

The Statistics/CMTS/Connections pane displays the CMTS connection statistics.


Related Topics ■ Overview of the JPS

Viewing JPS CMTS Locator Statistics with the C-Web Interface

Purpose View information about the recent performance statistics for the CMTS locator.

Action Click **JPS > Statistics > CMTS Locator**.

The Statistics/CMTS Locator pane displays the CMTS locator statistics.

Monitor	Configure	Diagnose	Manage	Logged in as: admin	Refresh	Preferences	About	Logout
CLI	JPS							
Component	Statistics / CMTS Locator							
Date								
Disk	JPS CMTS Locator							
Interfaces...	Number of lookups		0					
JPS	Number of no-match lookups		0					
NIC	Number of lookup errors		0					
NTP	Minimum lookup time (ms)		0					
Redirect Server	Average lookup time (last 100 lookups, ms)		0					
Route...	Maximum lookup time (ms)		0					
SAE								
Security								
System								
Copyright © 2007, Juniper Networks, Inc. All Rights Reserved . Trademark Notice . Privacy . 								

Related Topics ■ Overview of the JPS

Viewing JPS Message Handler Statistics with the C-Web Interface

Purpose View information about the JPS message handler.

Action Click **JPS > Statistics > Message Handler**.

The Statistics/Message Handler pane displays the JPS message handler statistics.

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 lists various components: CLI, Component, Date, Disk, Interfaces..., JPS (highlighted), NIC, NTP, Redirect Server, Route..., SAE, Security, and System. The main content area displays 'JPS Statistics / Message Handler' with a table of statistics:

JPS Message Handler	
Messages received	0
Message handled	0
Message dropped	0
Average non-decoding time in JPS (last 5000 messages, ms)	0
Throughput (last 60s, msgs/s)	0

The footer contains copyright information for Juniper Networks, Inc. (2007) and the Juniper logo.

Related Topics ■ Overview of the JPS

Viewing JPS Message Flow Statistics with the C-Web Interface

Purpose View information about JPS message flows.

Action 1. Click **JPS > Statistics > Message Handler > Message Flows**.

The Statistics/Message Handler/Message Flow pane appears.

The screenshot shows the Juniper C-Web Interface with the 'JPS Statistics / Message Handler / Message Flow' pane open. The left sidebar is the same as the previous screenshot, with 'JPS' highlighted. The main content area has a title bar 'JPS Statistics / Message Handler / Message Flow' and a form with an 'Id' input field. Below the input field are 'OK' and 'Reset' buttons. To the right of the input field is a text box explaining the ID filter:

Identifier for message flow.
Value: All or part of the message flow ID. If the message flow ID filter is not specified, all message flows are selected.
Default: No value

The footer contains copyright information for Juniper Networks, Inc. (2007) and the Juniper logo.

2. In the ID box, enter a message flow ID, or leave the box blank to display statistics for all message flows.
3. Click **OK**.

The Statistics/Message Handler/Message Flow pane displays the message flow statistics.

Related Topics ■ Overview of the JPS

Viewing JPS RKS Statistics with the C-Web Interface

Purpose View recent performance statistics for the policy server-to-record keeping server (RKS) interface.

Action Click **JPS > Statistics > RKS**.

The Statistics/RKS pane displays statistics for the policy server-to-RKS interface.

Monitor	Configure	Diagnose	Manage	Logged in as: admin	Refresh	Preferences	About	Logout
CLI				JPS				
Component				Statistics / RKS				
Date								
Disk				JPS Radius Plugin				
Interfaces...				Initial-Gate-Set observed	0			
JPS				Non-Initial-Gate-Set observed	0			
NIC				Gate-Set-Acks observed	0			
NTP				Gate-Set-Errs observed	0			
Redirect Server				Gate-Dels observed	0			
Route...				Gate-Del-Acks observed	0			
SAE				Gate-Del-Errs observed	0			
Security				Gate-Infos observed	0			
System				Gate-Info-Acks observed	0			
				Gate-Info-Errs observed	0			
				Gate-Report-State-Close observed	0			
				Gate-Report-State-Close-EGI-Status-Unknown observed	0			
				Gate-Report-State-Non-Close observed	0			
				Synch-Requests observed	0			
				Synch-Reports observed	0			
				Policy-Request events sent	0			
				Policy-Update events sent	0			
				Policy-Delete events sent	0			
				Time-Change events broadcast	0			
				Gate-Infos sent	0			
				Gate-Info-Acks received	0			

Related Topics ■ Overview of the JPS

Part 3

Managing Services on RADIUS Devices

- Managing Services on Third-Party Devices in the SRC Network on page 115
- Managing Services on RADIUS-Enabled Devices on page 123
- Monitoring the Diameter Server with the SRC CLI on page 141

Chapter 13

Managing Services on Third-Party Devices in the SRC Network

- Overview of CoA Script Service on page 115
- Configuring CoA Script Services on page 115
- Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 116
- Creating the CoA Script Service with the SRC CLI on page 116
- Configuring the CoA Script Service with the SRC CLI on page 117
- Parameters for Sample CoA Script Service on page 118
- Configuring Subscriptions to the CoA Script Service on page 119
- Example: Using the Sample CoA Script Service on page 119
- Defining RADIUS Attributes for CoA Requests with the API on page 120

Overview of CoA Script Service

The SAE can use change-of-authorization (CoA) messages to manage services for a specific subscriber session. The CoA script service allows the SAE to exchange CoA messages with third-party devices that do not support Common Open Policy Service (COPS) protocol to activate or deactivate services for specific subscriber sessions. When the SAE activates a CoA script service session, the session sends CoA messages to a RADIUS-enabled device. This method uses RADIUS attributes and RADIUS vendor-specific attributes (VSAs) to identify a subscriber session whose services are to be activated or deactivated.

Configuring CoA Script Services

To support CoA message exchange in an SRC network, configure a script service that can be activated on a third-party device. The script service defines the parameters needed to activate or deactivate services for a subscriber session, such as the address of the third-party device. This script service is activated for the subscriber session whose services are activated or deactivated. For detailed information about configuring script services, see Customizing Service Implementations.

When you use the CoA script service with third-party devices that do not notify the SAE about subscriber events, you must set up the Monitoring Agent application to handle RADIUS accounting request packets.

For information about configuring services on the third-party device, see the device's software documentation.

The tasks to set up the SRC software for CoA message exchange are:

- Configuring Monitoring Agent to Receive RADIUS Accounting Messages
- Creating the CoA Script Service with the SRC CLI
- Configuring the CoA Script Service with the SRC CLI
- Configuring Subscriptions to the CoA Script Service

The SRC software includes a sample script service that you can configure to exchange CoA messages with the third-party device. You can use the sample service definition and customize it for your environment by modifying the service substitutions. For information about the sample CoA script service, see *Example: Using the Sample CoA Script Service*.

Configuring Monitoring Agent to Receive RADIUS Accounting Messages

If you install the Monitoring Agent application on the same host as the RADIUS server, you must disable the `MonAgent.radius.server` property.

You can configure Monitoring Agent to act as a pseudo-RADIUS server that listens for RADIUS accounting packets sent to the RADIUS accounting port. To receive RADIUS packets from RADIUS clients:

- Make sure there is no other RADIUS server listening on the RADIUS accounting port, and enable the `MonAgent.radius.server` property.
- Configure the shared secret between the RADIUS server and the RADIUS client by specifying the `MonAgent.radius.secret. <IP address>` property.

For information about installing and using Monitoring Agent, see the *SRC Sample Applications Guide*.

Creating the CoA Script Service with the SRC CLI

To create the script service:

1. From configuration mode, enter the service configuration. In this sample procedure, the service is configured in the global service scope, and `CoAservice` is the name of the service.

```
user@host# edit services global service CoAservice
```

2. Configure the type of service.

```
[edit services global service CoAservice]
user@host# set type script
```

3. (Optional) Specify whether the service is visible only to administrators who have permission to see secret information.

```
[edit services global service CoAservice]
user@host# set secret
```

4. Configure URL as the type of script that the sample CoA script service uses.

```
[edit services global service CoAservice]
user@host# set script script-type url
```

5. Configure `net.juniper.smgmt.sae.coa.CoaService` as the name of the class that implements the script service.

```
[edit services global service CoAservice]
user@host# set script class-name net.juniper.smgmt.sae.coa.CoaService
```

6. Configure the URL of the script service or the path and filename of the service. Copy the `/lib/coa.jar` file used by the script service to a location that is accessible by a URL (such as an FTP or HTTP server). In this sample procedure, the `coa.jar` file was copied to the `/opt/UMC/sae/var/run` directory.

```
[edit services global service CoAservice]
user@host# set file file:///opt/UMC/sae/var/run/coa.jar
```

7. (Optional) Verify your configuration.

```
[edit services global service CoAservice]
user@host# show
type script;
status active;
available;
script {
  script-type url;
  class-name net.juniper.smgmt.sae.coa.CoaService;
  file file:///opt/UMC/sae/var/run/coa.jar;
}
```

After you create the script service, you need to configure parameters for the script service. For more information about configuring script services and parameters, see Overview of SRC Script Services.

Configuring the CoA Script Service with the SRC CLI

To configure the script service, you provide parameter substitutions with the values that are in the service definitions.

To configure parameters:

1. From configuration mode, enter the service parameter configuration. In this sample procedure, the service called `CoAservice` is configured in the global service scope.

```
user@host# edit services global service CoAservice parameter
```

2. (Optional) Configure actual values for other parameters.

```
[edit services global service CoA service parameter]
user@host# set substitution [ substitution... ]
```

The script file `/SDK/scriptServices/coa/ldif/BOD1M.ldif` in the `SDK+AppSupport+Demos+Samples.tar.gz` file provides parameters specified by the sample CoA script service. You can use the sample script service as a starting point. See [Parameters for Sample CoA Script Service](#).

Parameters for Sample CoA Script Service

Table 8 on page 118 lists the parameters specified by the sample CoA script service, which is the `/SDK/scriptServices/coa/ldif/BOD1M.ldif` file in the `SDK+AppSupport+Demos+Samples.tar.gz` file. You can use the sample script service as a starting point.

Table 8: Parameter Substitutions for CoA Services

Parameter Name	Description
dynClientIp	IP address of the third-party device.
dynClientPort	UDP port number of the third-party device.
dynSecret	Shared secret between RADIUS server and RADIUS client.
dynRetry	Number of retries for sending CoA messages when no RADIUS response is received. The retry interval is 3 seconds.

Table 8: Parameter Substitutions for CoA Services *(continued)*

Parameter Name	Description
dynConfig	<p>Content of service definition in the format <code><action> . <radiusAttributeName> = <pluginEventAttribute> \n</code></p> <ul style="list-style-type: none"> ■ action—Action that is executed on packet content (attribute): <ul style="list-style-type: none"> ■ start ■ stop ■ start-stop ■ radiusAttributeName—Valid RADIUS attribute specified as follows: <ul style="list-style-type: none"> ■ Standard RADIUS attribute name or number ■ Third-party VSA in the format vendor-specific. <vendor#> . <vsa#> .string ■ pluginEventAttribute—Valid expression in the format: <ul style="list-style-type: none"> ■ Python expression ■ <code><commandCode> <serviceName></code>; the entire expression must be enclosed in single quotation marks and you must use three backslashes (\) to escape the backslash that starts a <code><commandCode></code> For example: <code>\x0b</code> would be replaced by <code>\\x0b</code> ■ \n—New-line character included between the lines of a configuration containing multiple lines; the entire configuration must be enclosed in quotation marks. For example: <pre>start-stop.Acct-Session-Id = ifSessionId "start-stop.Acct-Session-Id = ifSessionId\nstart.vendor-specific.9.252.string = '\\x0bBOD1M'\nstop.vendor-specific.9.252.string = '\\x0cBOD1M'\n"</pre>

You can also configure dynamic RADIUS requests with the `sendDynamicRadius` method of the `ServiceSessionInfo` interface (see [Defining RADIUS Attributes for CoA Requests with the API](#)).

Configuring Subscriptions to the CoA Script Service

You need to configure subscriptions to the CoA script service. You can set up the subscriptions to activate immediately on login.

For more information, see [Adding Subscribers \(SRC CLI\)](#).

Example: Using the Sample CoA Script Service

To use the sample CoA script service provided:

1. Import the sample script service using an LDAP browser.

The `/SDK/scriptServices/coa/ldif/BOD1M.ldif` file (in the `SDK+AppSupport+Demos+Samples.tar.gz` file) is the sample service definition for exchanging CoA messages with a Cisco 10000 Series router.

2. Copy the `/lib/coa.jar` file used by the script service to a location that is accessible to the SAE by a URL, such as an FTP or HTTP server. If you do not have multiple SAEs, it can be convenient to copy the file to the `/var/run` directory in the SAE installation directory (`/opt/UMC/sae` by default).
3. Modify the service substitutions for your device.

You can make these substitutions by defining the parameter substitutions in the BOD1M service with the SRC CLI or by passing the values through the SAE core API.

For information about parameter substitutions, see *Configuring the CoA Script Service with the SRC CLI*. For information about passing the values through the SAE core API, see *Defining RADIUS Attributes for CoA Requests with the API*.

4. Configure a subscription to the BOD1M service that is activated on login.

For more information about subscriptions, see *Overview of Subscriptions*.

If you are modifying the sample application, add the `sae.jar` and `logger.jar` files to the classpath when you compile your application. These two files can be found in the `lib` directory of the SAE installation directory.

Defining RADIUS Attributes for CoA Requests with the API

The SRC software provides two ways to define RADIUS attributes for dynamic RADIUS authorization requests:

- Service definition (see *Configuring the CoA Script Service with the SRC CLI*)
- SAE core API



NOTE: Parameters set in the API override parameters set by the service definition.

To send dynamic RADIUS authorization requests with the SAE core API, the script service uses the `sendDynamicRadius` and `getRouterDynRadiusAddr` methods in the `ServiceSessionInfo` interface to provide the content of the RADIUS packet for the dynamic authorization request to the router that is attached to the service session.

For information about 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>.

For a sample implementation, see the following file in the `SDK+AppSupport+Demos+Samples.tar.gz` file:

SDK/scriptServices/coa/java/net/juniper/smg/scriptServices/coa/CoaService.java.

Chapter 14

Managing Services on RADIUS-Enabled Devices

- Overview of the IMS AAA Server Integration on page 123
- Managing Dynamic Services on page 124
- Configuring the IMS AAA Server on page 124
- Configuring the Diameter Application (SRC CLI) on page 125
- Configuring the NAS Groups (SRC CLI) on page 130
- Configuring the SAE to Manage AAA Devices on page 135
- Configuring AAA Policies (SRC CLI) on page 137

Overview of the IMS AAA Server Integration

When the Juniper IMS AAA Server is integrated in the SRC network, the SRC software can use the IMS AAA Server to dynamically manage services on RADIUS-enabled devices. The RADIUS capabilities of the IMS AAA Server allow the SRC software to be aware of the subscriber activity and to make dynamic RADIUS requests using these RADIUS features:

- Authentication, authorization, and accounting (AAA)
- Change-of-authorization (CoA) message
- Disconnect message (DM)

The SRC software communicates with the IMS AAA Server using the DIAMETER protocol. The IMS AAA Server uses RADIUS AAA messages to communicate with the RADIUS server and the Network Access Server (NAS). The IMS AAA Server acts as a proxy to convert Diameter messages to RADIUS messages and vice versa. The IMS AAA Server also performs conversion between Diameter attribute value peers (AVPs) and RADIUS attributes.

When integrated in the SRC network, the IMS AAA Server can provide:

- Device abstraction and shared secrets for the NAS device
- Accounting support for subscriber sessions and service sessions
- CoA and DM support
- Service parameter changes

Managing Dynamic Services

You can integrate the IMS AAA Server to support management of services on RADIUS-enabled devices in an SRC network. The IMS AAA Server will act as a proxy to intercept messages between the NAS device and the RADIUS server so that the SRC software does not need to know device details. You can configure the services, policies, and parameters with the SRC software independent of the NAS device. The SRC software communicates with the IMS AAA Server using Diameter messages to dynamically manage services for a subscriber session. The IMS AAA Server converts the Diameter messages to RADIUS messages and makes dynamic RADIUS requests to the NAS device.

The SRC software includes a Diameter server that forwards AAR, ACR, and STR messages from the IMS AAA Server to the AAA device driver in the SAE and that forwards PPR and ASR messages from the AAA device driver to the IMS AAA Server. These Diameter messages perform these functions:

- AAR—Attach subscriber to access network
- ACR—Provide accounting information
- ASR—Disconnect subscriber
- PPR—Start, modify, or stop service session; send message routing configuration
- STR—Detach subscriber from access network

You configure NAS groups and a AAA device driver for each NAS group hosted by the SAE. You also configure the services, policies, and parameters that the IMS AAA Server will use for service activation on the NAS device. You will need to provide specific information for the custom router template on the IMS AAA Server.

The custom router template (deviceModels.xml) lists the parameters needed for service activation on a NAS device (controlledDeviceModel element in the template). The IMS AAA Server has detailed knowledge about the specific NAS device so that it can use the services, policies, and parameters configured by the SRC software for managing services on the NAS device.

Tasks to set up the management of services on RADIUS-enabled devices are:

- Configure the IMS AAA Server.
- Configure the Diameter application.
- Configure the NAS groups.
- Configure the SAE to manage AAA devices.
- Configure AAA policies.

Configuring the IMS AAA Server

Tasks to set up the IMS AAA Server in the SRC network are:

- Configure the local elements for the IMS AAA Server.
 - Identification—IMS AAA Server

- Diameter configuration—Transport port protocol
- RADIUS configuration—AAA ports
- Configure the remote network elements.
 - The Diameter elements are for the C-series Controller and include: Origin host, IP address, port protocol, and function (SRC).
 - The RADIUS elements are for the NAS device and include: Name, IP address, and function.
- Review the custom router template for the NAS device. The service template element lists the parameters needed for service activation on the NAS device.

For information about configuring the IMS AAA Server with the IMS AAA Server Administrator, see the *IMS AAA Server Administration Guide*.

Configuring the Diameter Application (SRC CLI)

Tasks to configure the Diameter application are:

- Configuring the Diameter Application Properties on page 125
- Configuring the Diameter Client Properties on page 128
- Configuring the Diameter Server Properties on page 128
- Configuring Logging Destinations on page 129

Configuring the Diameter Application Properties

Use the following configuration statements to configure the properties for the Diameter application:

```
system diameter {
  java-heap-size java-heap-size;
  java-new-size java-new-size;
  java-garbage-collection-options java-garbage-collection-options;
  protocol [(tcp | sctp)...];
  local-address [local-address...];
  port port;
  origin-host origin-host;
  origin-realm origin-realm;
  active-peers;
  debug-mode;
  load-balancing-mode (failover | round-robin);
  transaction-processing-log (log-no-messages | log-severe-messages |
    log-normal-messages | log-debug-messages);
  packet-trace-log (log-no-messages | log-severe-messages | log-normal-messages |
    log-debug-messages);
  peer-state-machine-log (log-no-messages | log-severe-messages |
    log-normal-messages | log-debug-messages);
  configuration-log (log-no-messages | log-severe-messages | log-normal-messages |
    log-debug-messages);
}
```

To configure the Diameter application:

1. From configuration mode, access the configuration statement for the Diameter application.

```
user@host# edit system diameter
```

2. If you encounter problems caused by lack of memory, change the maximum memory size available to the JRE.

```
[edit system diameter]
user@host# set java-heap-size java-heap-size
```

3. Configure the amount of space available to the JRE when the Diameter server starts.

```
[edit system diameter]
user@host# set java-new-size java-new-size
```

4. Configure the garbage collection functionality of the Java Virtual Machine.

```
[edit system diameter]
user@host# set java-garbage-collection-options java-garbage-collection-options
```

5. Specify the protocol for the transport connection.

```
[edit system diameter]
user@host# set protocol [(tcp | sctp) ...]
```

6. (Optional) Specify the local IP addresses that remote peers can use to reach this server.

```
[edit system diameter]
user@host# set local-address [local-address ...]
```

7. (Optional) Specify the port for the server.

```
[edit system diameter]
user@host# set port port
```

8. (Optional) Specify the fully-qualified domain name used to identify this host to its Diameter peers.

```
[edit system diameter]
user@host# set origin-host origin-host
```

9. (Optional) Specify the DNS name used to identify this host to its Diameter peers.

```
[edit system diameter]
user@host# set origin-realm origin-realm
```

10. (Optional) Specify whether the peer connection is in active mode and automatically starts communicating with the IMS AAA Server.

```
[edit system diameter]
user@host# set active-peers
```

11. (Optional) Specify whether the peer connection is in debug mode.

```
[edit system diameter]
user@host# set debug-mode
```

12. (Optional) Configure the load-balancing mode for peer selection when forwarding a request message.

```
[edit system diameter]
user@host# set load-balancing-mode (failover | round-robin)
```

13. (Optional) Configure the log level for the transaction processing log.

```
[edit system diameter]
user@host# set transaction-processing-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

14. (Optional) Configure the log level for the packet tracing log.

```
[edit system diameter]
user@host# set packet-trace-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

15. (Optional) Configure the log level for the peer state machine log.

```
[edit system diameter]
user@host# set peer-state-machine-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.

- `log-normal-messages`—Log only normal messages.
- `log-debug-messages`—Log only debug messages.

16. (Optional) Configure the log level for the configuration log.

```
[edit system diameter]
user@host# set configuration-log log-level
```

where *log-level* is one of the following:

- `log-no-messages`—Do not log any messages.
- `log-severe-messages`—Log only severe messages.
- `log-normal-messages`—Log only normal messages.
- `log-debug-messages`—Log only debug messages.

Configuring the Diameter Client Properties

Use the following statements to configure the properties for the Diameter client:

```
system diameter client {
  threads threads;
  keep-alive-time keep-alive-time;
}
```

To configure the Diameter client properties:

1. From configuration mode, access the configuration statement for the Diameter client.

```
user@host# edit system diameter client
```

2. (Optional) Specify the minimum number of threads to use.

```
[edit system diameter client]
user@host# set threads threads
```

3. (Optional) Specify the time to wait for new commands.

```
[edit system diameter client]
user@host# set keep-alive-time keep-alive-time
```

Configuring the Diameter Server Properties

Use the following statements to configure the properties for the Diameter server:

```
system diameter server {
  threads threads;
  keep-alive-time keep-alive-time;
}
```

To configure the Diameter server properties:

1. From configuration mode, access the configuration statement for the Diameter server.

```
user@host# edit system diameter server
```

2. (Optional) Specify the minimum number of threads to use.

```
[edit system diameter server]
user@host# set threads threads
```

3. (Optional) Specify the time to wait for new commands.

```
[edit system diameter server]
user@host# set keep-alive-time keep-alive-time
```

Configuring Logging Destinations

Use the following configuration statements to configure logging destinations for Diameter:

```
system diameter logger name ...

system diameter logger name file {
  filter filter;
  filename filename;
  rollover-filename rollover-filename;
  maximum-file-size maximum-file-size;
}
```

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.

```
user@host# edit system diameter logger name file
```

2. Specify the properties for the logging destination.

```
[edit system diameter logger name file]
user@host# set ?
```

For more information about configuring properties for the logging destination, see Configuring Logging Destinations to Store Messages in a File.

Configuring the NAS Groups (SRC CLI)

Tasks to configure the NAS groups are:

- Configuring NAS Groups on page 130
- Configuring Diameter Peers on page 131
- Classifying Interfaces on page 132
- Selecting Routes on page 133

Configuring NAS Groups

Use the following configuration statements to configure the NAS groups:

```
shared network nas-group name {
  hosted-by [hosted-by...];
  function (aaa);
  scope [scope...];
  default-peer default-peer;
  update-grace-period update-grace-period;
  initial-ppr-delay initial-ppr-delay;
}
```

To configure the group of peers:

1. From configuration mode, access the configuration statements for the NAS group.

```
user@host# edit shared network nas-group name
```

2. Specify the hosts that instantiate this peer group. If the peer group is a AAA peer group, the SAEs on the listed hosts will create device drivers for this peer group.

```
[edit shared network nas-group name]
user@host# set hosted-by [hosted-by...]
```

3. Specify the functional interface of the peer group.

```
[edit shared network nas-group name]
user@host# set function aaa
```

4. (Optional) Specify the service scopes available to subscribers connected to this NAS group.

```
[edit shared network nas-group name]
user@host# set scope [scope...]
```

5. (Optional) Specify the default peer.

```
[edit shared network nas-group name]
user@host# set default-peer default-peer
```

6. (Optional) Specify the grace period for interim updates.

```
[edit shared network nas-group name]
user@host# set update-grace-period update-grace-period
```

7. (Optional) Specify the delay for sending initial Push-Profile-Requests (PPRs) to install policies.

```
[edit shared network nas-group name]
user@host# set initial-ppr-delay initial-ppr-delay
```

Configuring Diameter Peers

Use the following configuration statements to configure the Diameter peers:

```
shared network nas-group name peer name {
  protocol [(tcp | sctp)...];
  address [address...];
  local-address local-address;
  connect-timeout connect-timeout;
  watchdog-timeout watchdog-timeout;
  state-machine-timeout state-machine-timeout;
  reconnect-timeout reconnect-timeout;
  port port;
  origin-host origin-host;
  incoming-queue-limit incoming-queue-limit;
  active-peer;
}
```

To configure the Diameter peer in the NAS group:

1. From configuration mode, access the configuration statements for the peer.

```
user@host# edit shared network nas-group name peer name
```

The peer name must be unique in the NAS group.

2. Specify the protocol for the transport connection.

```
[edit shared network nas-group name peer name]
user@host# set protocol [(tcp | sctp)...]
```

3. Specify the addresses of the remote peer.

```
[edit shared network nas-group name peer name]
user@host# set address [address...]
```

4. (Optional) Specify the local address of the peer.

```
[edit shared network nas-group name peer name]
user@host# set local-address local-address
```

5. (Optional) Specify the maximum amount of time to respond to a connection request.

```
[edit shared network nas-group name peer name]
user@host# set connect-timeout connect-timeout
```

6. (Optional) Specify the watchdog timeout used for the connection to the remote peer.

```
[edit shared network nas-group name peer name]
user@host# set watchdog-timeout watchdog-timeout
```

7. (Optional) Specify the Diameter state machine timeout.

```
[edit shared network nas-group name peer name]
user@host# set state-machine-timeout state-machine-timeout
```

8. (Optional) Specify the time interval between connection attempts when the peer is in the disconnected state.

```
[edit shared network nas-group name peer name]
user@host# set reconnect-timeout reconnect-timeout
```

9. (Optional) Specify the port for the client.

```
[edit shared network nas-group name peer name]
user@host# set port port
```

10. (Optional) Specify the identifier for the endpoint that the peer presents during connection establishment.

```
[edit shared network nas-group name peer name]
user@host# set origin-host origin-host
```

11. (Optional) Specify the number of messages allowed on the incoming message queue for a peer.

```
[edit shared network nas-group name peer name]
user@host# set incoming-queue-limit incoming-queue-limit
```

12. (Optional) Specify whether the peer connection is in active mode and automatically starts communicating with the IMS AAA Server.

```
[edit shared network nas-group name peer name]
user@host# set active-peer
```

Classifying Interfaces

Use the following configuration statements to define interface classification scripts:

```
shared network nas-group name interface-classifier rule name {
  script script;
}
```

```
shared network nas-group name interface-classifier rule name {
    target target;
}
```

```
shared network nas-group name interface-classifier rule name condition name ...
```

A classification script can contain either a target and a condition or a script. If you do not define a script, the classifier must have both a target and a condition.

To define interface classification scripts:

1. From configuration mode, enter the interface classifier configuration for a NAS group.

```
user@host# edit shared network nas-group name interface-classifier
```

2. Create a rule for the classifier. You can create multiple rules for the classifier.

```
[edit shared network nas-group name interface-classifier]
user@host# edit rule name
```

3. Configure either a target or a script for the rule.

```
[edit shared network nas-group name interface-classifier rule name]
user@host# set script script
```

OR

```
[edit shared network nas-group name interface-classifier rule name]
user@host# set target target
```

4. If you configured a target for the rule, you must configure a match condition for the rule. You can create multiple conditions for the rule. See Interface Classification Conditions.

```
[edit shared network nas-group name interface-classifier rule name]
user@host# set condition name
```

Selecting Routes

Use the following configuration statements to configure the route for messages:

```
shared network nas-group name routes name term name {
    precedence precedence;
}
```

```
shared network nas-group name routes name {
    transaction-variable (request-packet | user-name | realm);
    dictionary-attribute (user-name | user-password | chap-password | nas-ip-address |
        nas-port | service-type | framed-protocol | framed-ip-address | framed-ip-netmask
        | framed-mtu | framed-compression | login-ip-host | callback-number | state |
        vendor-specific | called-station-id | calling-station-id | nas-identifier | login-lat-service
```

```

        | login-lat-node | login-lat-group | chap-challenge | nas-port-type | port-limit |
        login-lat-port);
operator (equals | not_equal | present | not_present | prefix | suffix | range);
value value;
low low;
high high;
}

```

To configure route selection for messages from the IMS AAA Server:

1. From configuration mode, access the configuration statements for route selection.

```
user@host# edit shared network nas-group name routes name
```

2. (Optional) Specify the order by which the route is selected. The route that meets all the matching criteria and has the lowest precedence is selected first. Routes without the precedence defined are considered after those that have the precedence defined. The route with precedence of -1 is the default route. The default route is considered after all the other routes, and only one default route can be defined.

```

[edit shared network nas-group name routes name]
user@host# set precedence precedence

```

3. From configuration mode, access the configuration statements for route selection criteria.

```
user@host# edit shared network nas-group name routes name term name
```

All the criteria must match for this route to be selected.

4. Specify the name of the transaction variable used as the matching criterion.

```

[edit shared network nas-group name routes name term name]
user@host# set transaction-variable (request-packet | user-name | realm)

```

5. (Optional) Specify the name of the dictionary attribute contained in the attribute store. Only applicable if the transaction variable is request-packet.

```

[edit shared network nas-group name routes name term name]
user@host# set dictionary-attribute (user-name | user-password | chap-password
| nas-ip-address | nas-port | service-type | framed-protocol | framed-ip-address
| framed-ip-netmask | framed-mtu | framed-compression | login-ip-host |
callback-number | state | vendor-specific | called-station-id | calling-station-id
| nas-identifier | login-lat-service | login-lat-node | login-lat-group |
chap-challenge | nas-port-type | port-limit | login-lat-port)

```

6. Specify the operator for criterion matching.

```

[edit shared network nas-group name routes name term name]
user@host# set operator (equals | not_equal | present | not_present | prefix |
suffix | range)

```

7. (Optional) Specify the value to be matched by the target.

```
[edit shared network nas-group name routes name term name]
user@host# set value value
```

8. (Optional) Specify the low end of the range criterion.

```
[edit shared network nas-group name routes name term name]
user@host# set low low
```

9. (Optional) Specify the high end of the range criterion.

```
[edit shared network nas-group name routes name term name]
user@host# set high high
```

Configuring the SAE to Manage AAA Devices

Use the following configuration statements to configure the AAA device driver:

```
shared sae configuration driver aaa {
  sae-community-manager sae-community-manager;
  origin-host origin-host;
  origin-realm origin-realm;
  keep-alive-timeout keep-alive-timeout;
  registry-retry-interval registry-retry-interval;
  reply-timeout reply-timeout;
  sequential-message-timeout sequential-message-timeout;
  transient-session-timeout transient-session-timeout;
  max-update-interval max-update-interval;
  update-grace-period update-grace-period;
  resume-unrecovered;
  thread-pool-size thread-pool-size;
  thread-idle-timeout thread-idle-timeout;
}
```

To configure the AAA device driver:

1. From configuration mode, access the configuration statements for the AAA device driver.

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

2. Specify the name of the community manager.

```
[edit shared sae configuration driver aaa]
user@host# set sae-community-manager sae-community-manager
```

3. (Optional) Specify the fully qualified domain name used to identify this host.

```
[edit shared sae configuration driver aaa]
user@host# set origin-host origin-host
```

4. (Optional) Specify the DNS name of the machine used to identify this host.

```
[edit shared sae configuration driver aaa]
user@host# set origin-realm origin-realm
```

5. (Optional) Specify the keepalive timeout before the registry to a Diameter server expires.

```
[edit shared sae configuration driver aaa]
user@host# set keep-alive-timeout keep-alive-timeout
```

6. (Optional) Specify the interval between retrying a failed registry to a Diameter server.

```
[edit shared sae configuration driver aaa]
user@host# set registry-retry-interval registry-retry-interval
```

7. (Optional) Specify the timeout before a request sent to a Diameter server expires.

```
[edit shared sae configuration driver aaa]
user@host# set reply-timeout reply-timeout
```

8. (Optional) Specify the timeout before an expected message expires.

```
[edit shared sae configuration driver aaa]
user@host# set sequential-message-timeout sequential-message-timeout
```

9. (Optional) Specify the timeout before a temporary session expires.

```
[edit shared sae configuration driver aaa]
user@host# set transient-session-timeout transient-session-timeout
```

10. (Optional) Specify the maximum interval between interim updates for a subscriber session.

```
[edit shared sae configuration driver aaa]
user@host# set max-update-interval max-update-interval
```

11. (Optional) Specify the grace period in which to expect an interim update for a subscriber session.

```
[edit shared sae configuration driver aaa]
user@host# set update-grace-period update-grace-period
```

12. (Optional) Specify whether to resume a subscriber session that has failed to recover from a failover.

```
[edit shared sae configuration driver aaa]
user@host# set resume-unrecovered
```

13. (Optional) Specify the number of working threads that process requests.

```
[edit shared sae configuration driver aaa]
user@host# set thread-pool-size thread-pool-size
```

14. (Optional) Specify the timeout for stopping working threads after they become idle.

```
[edit shared sae configuration driver aaa]
user@host# set thread-idle-timeout thread-idle-timeout
```

15. (Optional) Configure the session store parameters for the AAA device driver.

From configuration mode, access the configuration statement that configures the session store for the AAA device driver.

```
user@host# edit shared sae configuration driver aaa session-store
```

For more information about configuring session store parameters, see Configuring the Session Store Feature.

Configuring AAA Policies (SRC CLI)

Tasks to configure AAA policies are:

- Configuring AAA Policy Lists on page 137
- Configuring AAA Policy Rules on page 137
- Configuring Template Activation Actions on page 138

Configuring AAA Policy Lists

To configure AAA policy lists:

1. From configuration mode, create a policy list. For example, to create a policy list called l1 within a policy group called tiered_aaa:

```
user@host# edit policies group tiered_aaa list l1
```

2. Specify the type of policy list.

```
[edit policies group tiered_aaa list l1]
user@host# set role aaa
```

3. Specify where the policy is applied on the device.

```
[edit policies group tiered_aaa list l1]
user@host# set applicability both
```

Configuring AAA Policy Rules

To configure AAA policy rules:

1. From configuration mode, create a policy rule inside a policy list that has already been created and configured. For example, to create a policy rule called r1 within policy list l1:

```
user@host# edit policies group tiered_aaa list l1 rule r1
```

2. Specify the type of policy rule.

```
[edit policies group tiered_aaa list l1 rule r1]
user@host# set type aaa
```

Configuring Template Activation Actions

Use this action to activate templates for RADIUS-enabled devices. You can configure template activation actions for AAA policy rules.

The template name and parameters are listed in the custom router template on the IMS AAA Server.

Use the following configuration statements to configure a template activation action:

```

policies group name list name rule name template-activation name {
    template-name template-name;
    description description;
}

policies group name list name rule name template-activation name variables name {
    value value;
    type type;
}

```

To configure a template activation action:

1. From configuration mode, enter the template activation action configuration.
For example, in this procedure, *ta* is the name of the template activation action.

```
user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta
```

2. Enter the template name to activate.

```

[edit policies group tiered_aaa list l1 rule r1 template-activation ta]
user@host# set template-name template-name

```

3. (Optional) Enter a description for the template activation action.

```

[edit policies group tiered_aaa list l1 rule r1 template-activation ta]
user@host# set description description

```

4. From configuration mode, enter the parameters used by the template.

```
user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta  
variables name
```

For example:

```
user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta  
variables upstreamBandwidth
```

5. (Optional) Configure the value for the variable.

```

[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables name]
user@host# set value value

```

For example:

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables
upstreamBandwidth]
user@host# set value rateParameter
```

6. (Optional) Configure the variable type. Variable types are mapped to parameter types.

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables name]
user@host# set type type
```

For example:

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables
upstreamBandwidth]
user@host# set type rate
```


Chapter 15

Monitoring the Diameter Server with the SRC CLI

- SRC CLI Commands to Monitor the Diameter Server on page 141
- Viewing Statistics for the Diameter Server (SRC CLI) on page 142
- Viewing Message Handler Information for the Diameter Server (SRC CLI) on page 142
- Viewing Server Process Information for the Diameter Server (SRC CLI) on page 142
- Viewing Information About Diameter Server Requests (SRC CLI) on page 142
- Viewing Diameter Server State (SRC CLI) on page 142

SRC CLI Commands to Monitor the Diameter Server

You can view statistics and status for the Diameter server. Table 9 on page 141 lists the commands you use to monitor the Diameter server

Table 9: Commands to Monitor the Diameter Server

Command	Output Displayed
show diameter statistics	Information about the server process and the current state of the Diameter server.
show diameter statistics message-handler	Information about the Diameter server message handler.
show diameter statistics message-handler message-flow	Information about the Diameter server message flows.
show diameter statistics process	Information about the Diameter server process.
show diameter statistics requests	Information about the Diameter server requests.
show diameter status	Status of the Diameter server.
show diameter status clients	Status of the Diameter clients.
show diameter status peers	Status of the Diameter peers.

Viewing Statistics for the Diameter Server (SRC CLI)

Purpose View information about the server process and the state of the Diameter server.

Action To display information about the server process and the state of the Diameter server:

```
user@host > show diameter statistics
```

Viewing Message Handler Information for the Diameter Server (SRC CLI)

Purpose View information about the message handler and message flows for the Diameter server.

Action To display information about the message handler for the Diameter server:

```
user@host > show diameter statistics message-handler
```

To display information about message flows for the Diameter server:

```
user@host > show diameter statistics message-handler message-flow
```

To display information about a specific message flow:

```
user@host > show diameter statistics message-handler message-flow id id
```

Viewing Server Process Information for the Diameter Server (SRC CLI)

Purpose View information about the server process.

Action Purpose View information about the server process. Action To display about the server process:

```
user@host > show diameter statistics process
```

Viewing Information About Diameter Server Requests (SRC CLI)

Purpose View information about Diameter server requests.

Action To display information about Diameter server requests:

```
user@host > show diameter statistics requests
```

Viewing Diameter Server State (SRC CLI)

Purpose

Action To display information about the state of the Diameter server:

```
user@host > show diameter status
```

To display information about the Diameter clients:

```
user@host > show diameter status clients
```

To display information about a specific client:

```
user@host > show diameter status clients client-name client-name
```

To display information about the Diameter peers:

```
user@host > show diameter status peers
```

To display information about a specific peer:

```
user@host > show diameter status peers peer-name peer-name
```


Part 4

Providing Services in IMS Networks

- Providing Services in IMS Networks on page 147
- Providing Services in IMS Networks with the SRC CLI on page 155

Chapter 16

Providing Services in IMS Networks

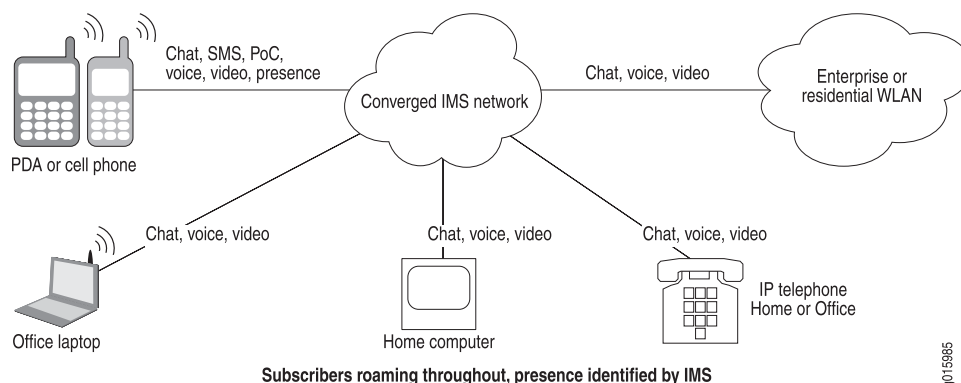
- Overview of an IMS Environment on page 147
- IMS and ETSI References on page 148
- IMS Layers on page 149
- ETSI-TISPAN Architecture on page 150
- SRC Software in the ETSI-TISPAN Architecture on page 152
- SRC Software in the IMS Environment on page 153

Overview of an IMS Environment

IP multimedia subsystem (IMS) is a flexible network architecture that allows providers to introduce rich multimedia services across both next-generation packet-switched and traditional circuit-switched networks. It uses open interfaces and functional components that can be assembled flexibly to support real-time interactive services and applications.

Third Generation Partnership Project (3GPP) developed IMS to provide a standards-based architecture for mobile carriers to migrate to their next-generation networks that will support applications that combine voice, video, and data functionality. The European Telecommunications Standards Institute (ETSI) created Telecommunications and Internet Converged Services and Protocols for Advanced Networks (TISPAN) to extend IMS support to fixed-line carriers. This extension is commonly called fixed mobile convergence (FMC). IMS/FMC allows subscribers to access any network (wireless or fixed) from any device (computer, PDA, or cell phone) and be able to move seamlessly from one network to another.

Figure 16 on page 148 shows, at a high level, a converged IMS network that manages and controls the movement of subscribers between fixed and wireless networks.

Figure 16: A Simplified IMS Converged Network (Service Focus)

By itself, IMS does not specify new services; rather, it provides a framework for network operators to build and launch their services regardless of access method. The IMS architecture simplifies network operations and allows providers to focus on service introduction and business opportunities. For example, an IMS architecture could allow fixed and mobile users to communicate using voice, video, chat, and online gaming, and to take advantage of functionality such as Push-to-Talk over Cellular (PoC; the ability to quickly arrange meetings through a walkie-talkie mechanism), instant messaging, and presence (whether and how a subscriber is available, and how the subscriber wants to be contacted).

IMS and ETSI References

For more information about IMS and TISPAN, consult the following specifications:

- ETSI ES 283 026 V0.0.7 (2005-10) *Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Resource and Admission Control; Protocol for QoS reservation information exchange between the Service Policy Decision Function (SPDF) and the Access-Resource and Admission Control Function (A-RACF) in the Resource and Protocol specification.*
- ETSI TS 183 017 V.0.0.8 (2005-10) *Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Resource and Admission Control: DIAMETER protocol for session based policy set-up information exchange between the Application Function (AF) and the Service Policy Decision Function (SPDF); Protocol specification.*
- ETSI ES 283 034 V0.0.5 (2005-10) *Telecommunications and Internet converged Services and Protocols for Advanced Networks (TISPAN); Network Attachment Sub-System (NASS); e4 interface based on the DIAMETER protocol.*

Abbreviations

Table 10 on page 149 identifies abbreviations used in the IMS and ETSI-TISPAN environments.

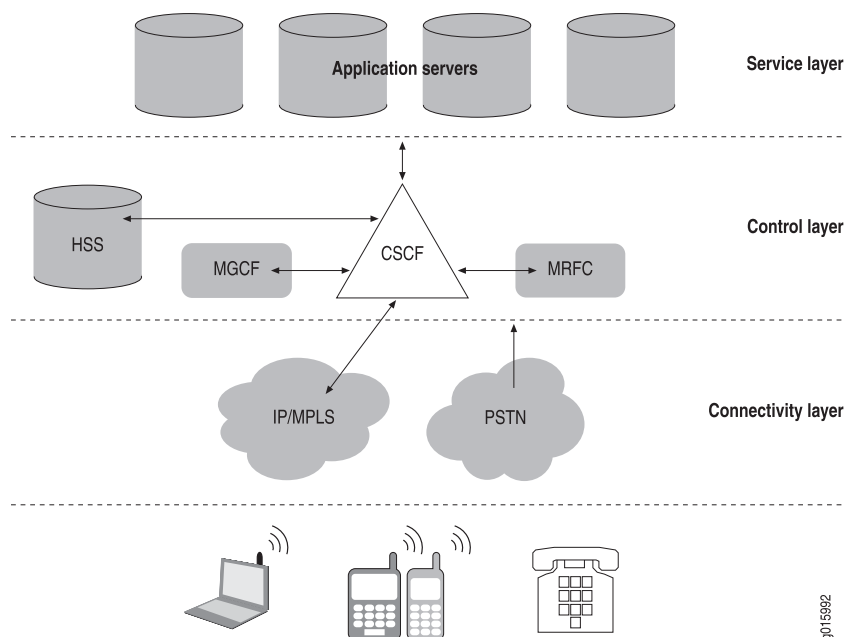
Table 10: Abbreviations in the IMS and ETSI-TISPAN Environments

Abbreviation	Description
3GPP	3rd Generation Partnership Project, which developed the IMS specifications.
A-RACF	Access-resource and admission control function. Provides admission control and network policy assembly.
AVP	Attribute value pair
BGF	Border gateway function
ETSI	European Telecommunications Standards Institute
FMC	Fixed mobile convergence
IMS	IP multimedia subsystem
NGN	Next-generation network
RACS	Resource and admission control subsystem. Consists of the A-RACF and the SPDF.
RCEF	Resource control enforcement function
SPDF	Service policy decision function. The SPDF coordinates the resource reservations requests that it receives from the application function.
TISPAN	Telecommunications and Internet Converged Services and Protocols for Advanced Networks

IMS Layers

The IMS specifications define functions to handle the signaling and subscriber traffic for multimedia applications. The functions are separated into logical layers, and many of the specified functions often reside in a single platform. Vendors have the flexibility to implement IMS functions in consolidated ways, and it is natural that platforms such as softswitches will combine many logically separate IMS call-processing functions, and that routers will take on some of the session-enforcement and gateway functionality in IMS.

The three layers are the service layer, the control layer, and the transport layer. Figure 17 on page 150 shows a high-level view of the IMS architecture.

Figure 17: High-Level View of the IMS Architecture

- **Service layer**—Hosts application and content services, including application servers and Web servers. It also includes generic service enablers that manage service elements such as user groups and presence. These service elements connect to subscribers through the control plane. The application layer supports most of the multimedia applications or application enablers, such as presence and location of the subscriber.
- **Control layer**—Makes the policy decisions that are enforced in the transport layer. This layer provides session control and management, and is responsible for setting up and taking down packet sessions. It also contains information about subscriber authentication, service authorization, and location.
- **Connectivity layer**—Supports the core network architecture of the General Packet Radio Service (GPRS), which consists of support nodes for data services. This layer is where routers, switches, firewalls, and optical transport reside, along with gateways that translate protocols between packet- and circuit-based traffic.

Signaling Protocol

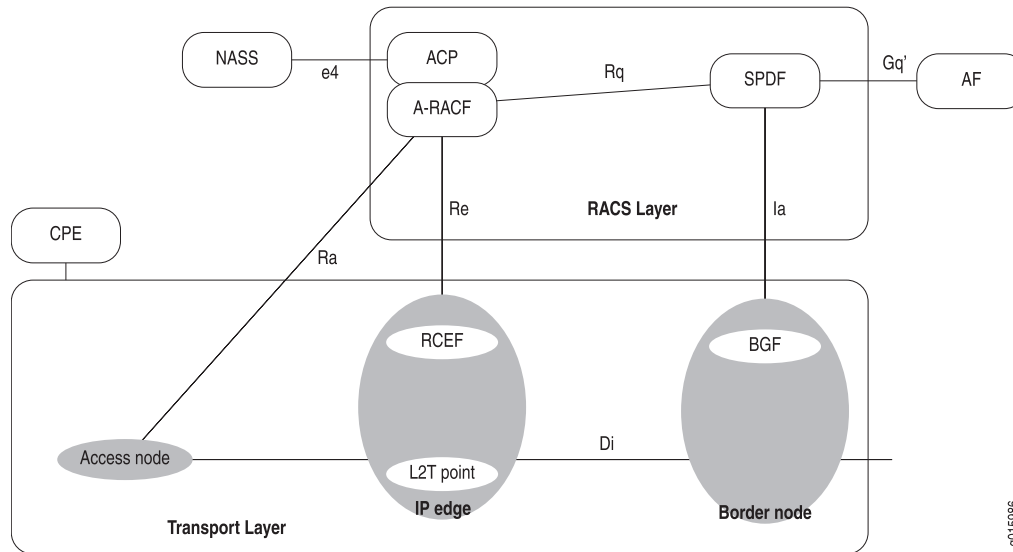
Session Initiation Protocol (SIP) is the main signaling protocol in IMS. SIP is the proposed standard for multimedia communication between subscribers interacting with voice, video, and instant messaging. In IMS, the use of SIP facilitates interconnectivity between fixed and mobile networks.

ETSI-TISPAN Architecture

TISPAN is an extension to the IMS architecture developed by ETSI to fit the specific requirements of fixed-line providers.

Figure 18 on page 151 shows a high-level view of the TISpan architecture.

Figure 18: High-Level View of the ETSI-TISpan Architecture



RACS Layer

The RACS layer is the TISpan next-generation network subsystem that is responsible for elements of policing control, including resource reservation and admission control in the access and aggregation networks. The RACS layer also includes support for NAT in the access, aggregation, and core networks required to support end-to-end application-initiated sessions.

The RACS provides policy-based transport control services to applications. These services enable applications to request and reserve transport resources from transport resources from the transport networks within the scope of the RACS.

Rq Interface

The Rq interface is the interface between the SPDF and the A-RACF. The SPDF issues requests for resources in the access network through the Rq interface. These requests indicate IP QoS characteristics. The A-RACF uses the IP QoS information to perform admission control and indicates to the SPDF through the Rq interface its admission control decisions.

SPDF

The SPDF is a functional element that coordinates the resource reservations requests that it receives from the application function (the application-level controller, such as a SIP server). The SPDF performs the following functions:

- Determines whether the request information received from the application function is consistent with the policy rules defined in the SPDF.
- Authorizes the requested resources for the application function session. The SPDF uses the request information received from the application function to calculate the proper authorization (that is, to authorize certain media components).
- Provides the location of the BGF and/or the A-RACF device, in accordance with the required transport capabilities.
- Requests resources of the A-RACF.
- Requests services from the BGF.
- Hides the details of the RACS and the core transport layer from the control architecture.
- Provides resource mediation by mapping requests from application functions toward an appropriate A-RACF and/or BGF.

A-RACF

The A-RACF is a functional element that provides admission control and network policy assembly.

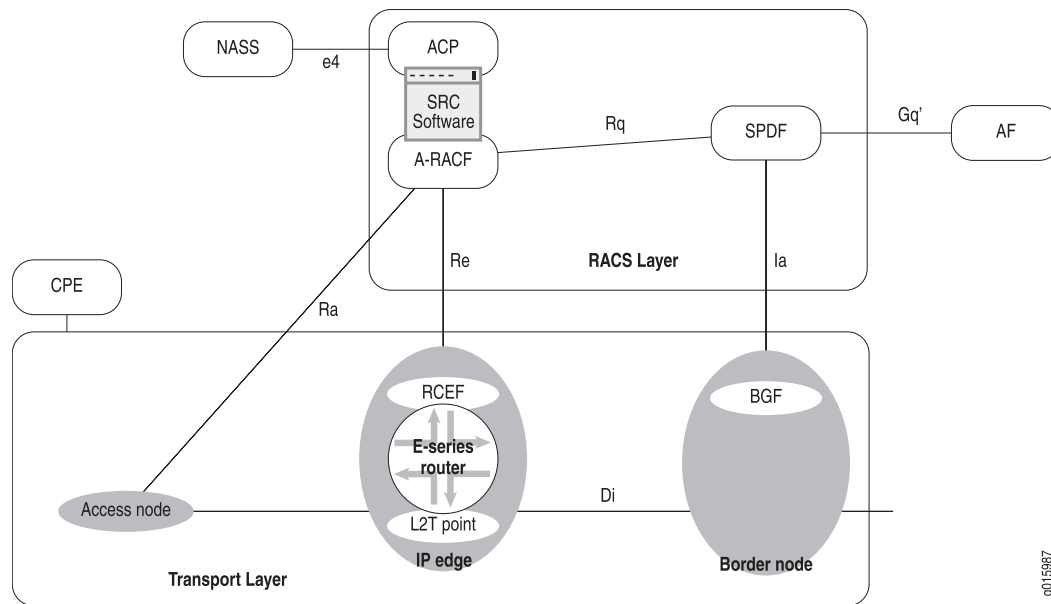
For admission control, the A-RACF receives requests for QoS resources from the SPDF and uses the QoS information received to perform admission control. It then indicates to the SPDF whether or not a request for resources is granted.

Access network policies are a set of rules that specify the policies that should be applied to an access line. For network policy assembly, the A-RACF:

- Ensures that requests from the SPDF match the access policies because multiple SPDFs can request resources from the A-RACF.
- Combines the requests from the SPDFs that have requested resources and ensures that the total of the requests match the capabilities of the access line.

SRC Software in the ETSI-TISPAN Architecture

Figure 19 on page 153 shows the SRC software in the ETSI-TISPAN architecture.

Figure 19: SRC Software in the ETSI-TISPAN Architecture

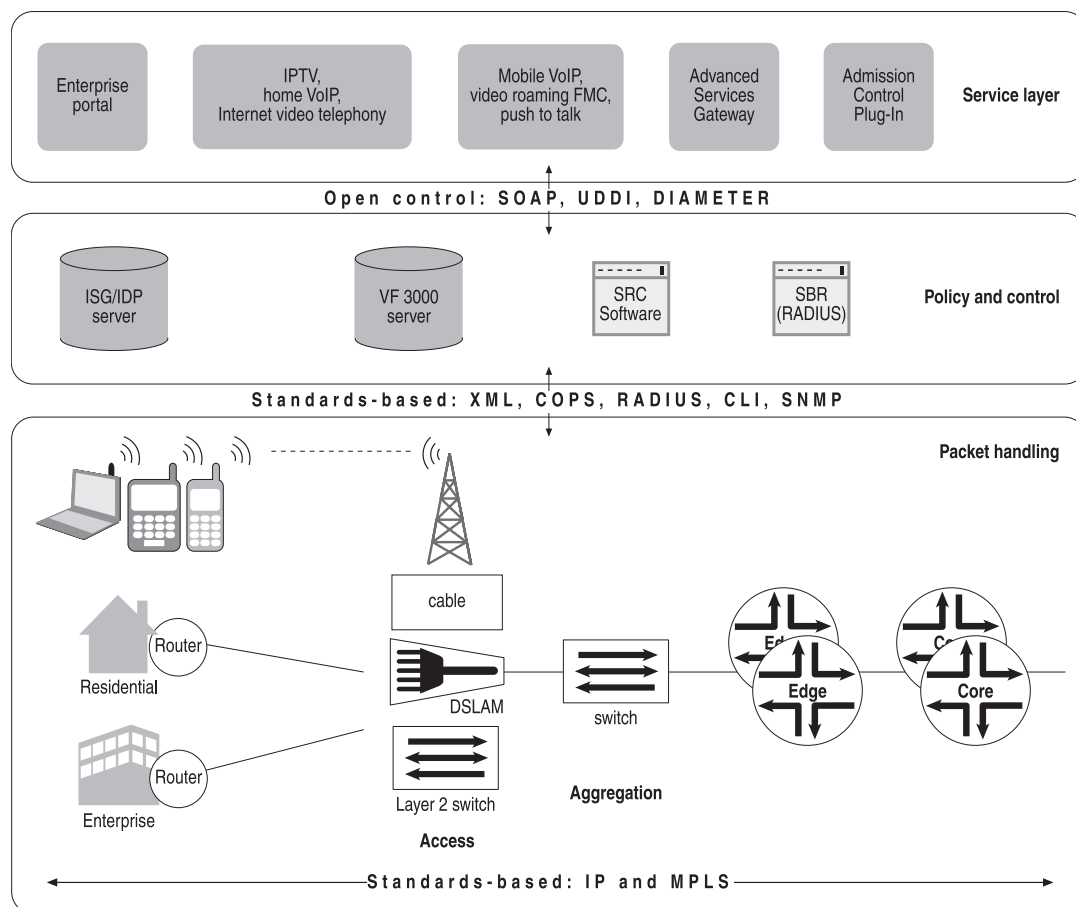
The SAE provides the A-RACF functionality, and the SRC software provides a northbound Rq interface from the A-RACS to the SPDF. This interface is equivalent to the Rq interface defined in the ETSI-TISPAN release 1 architecture. It is a DIAMETER protocol-based interface that allows the SRC software to integrate with services found on the application layer of IMS.

The SRC software uses its COPS and BEEP interfaces as the Re interface to Juniper Networks routers.

SRC Software in the IMS Environment

Figure 20 on page 154 shows the Juniper Networks layered IMS architecture.

The northbound Rq interface of the policy and control layer allows integration with SRC applications, such as the portals, the Advanced Services Gateway, and the Admission Control Plug-In.

Figure 20: Juniper Networks IMS Architecture

Chapter 17

Providing Services in IMS Networks with the SRC CLI

- Configuration Statements for IMS Support on page 155
- Configuring the IMS Software on page 156
- Configuring Initial Properties for IMS on page 157
- Configuring Directory Connection Properties for IMS on page 158
- Configuring Initial Directory Eventing Properties for IMS on page 158
- Configuring the Local Diameter Peer on page 159
- Configuring the Remote Diameter Peer on page 160
- Configuring Logging Destinations to Store Messages in a File on page 161
- Configuring Logging Destinations to Send Messages to the System Logging Facility on page 162
- Configuring the Subscriber Type on page 162
- Configuring a NIC Proxy for IMS on page 163
- Managing IMS on page 169
- Monitoring IMS with the SRC CLI on page 170
- Monitoring IMS with the C-Web Interface on page 171
- Example: Configuring JUNOS Policies for IMS with the SRC CLI on page 172

Configuration Statements for IMS Support

Use the following configuration statements to configure IMS support at the [edit] hierarchy level.

```
slot number ims aracf-rq {  
    protocol protocol;  
    port port;  
    address address;  
    origin-host origin-host;  
    origin-realm origin-realm;  
}  
slot number ims aracf-rq peer primary-spdf {  
    address address;  
    origin-host origin-host;
```

```

}
slot number ims initial {
    static-dn static-dn;
    dynamic-dn dynamic-dn;
}
slot number ims 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 ims initial directory-eventing {
    eventing;
    signature-dn signature-dn;
    polling-interval polling-interval;
    event-base-dn event-base-dn;
    dispatcher-pool-size dispatcher-pool-size;
}
slot number ims logger name ...
slot number ims logger name file {
    filter filter;
    filename filename;
    rollover-filename rollover-filename;
    maximum-file-size maximum-file-size;
}
slot number ims logger name syslog {
    filter filter;
    host host;
    facility facility;
    format format;
}

```

Related Topics ■ For more information about the configuration statements, see the *SRC-PE CLI Command Reference*.

Configuring the IMS Software

To configure the IMS software:

1. Configure initial properties, including the connection to the directory and directory monitoring properties.

See Configuring Initial Properties for IMS.

See Configuring Directory Connection Properties for IMS.

See Configuring Initial Directory Eventing Properties for IMS.

2. Configure the local and remote Diameter peers.

See Configuring the Local Diameter Peer.

See Configuring the Remote Diameter Peer.

3. Configure logging destinations.

See Configuring System Logging with SRC CLI and Configuring a Component to Store Log Messages in a File with SRC CLI.

4. Configure subscriber types.

See Configuring the Subscriber Type .

5. Configure the NIC proxies.

See Configuring a NIC Proxy for IMS .

6. Start the IMS process to provide the A-RACF Rq interface.

See Starting the IMS Process .

You must restart the IMS process after you commit a configuration change. To restart IMS, see Restarting the IMS Process.

Configuring Initial Properties for IMS

Use the following configuration statements to configure initial properties for IMS:

```
slot number ims 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 ims initial
```

2. Specify the properties for IMS.

```
[edit slot 0 ims initial]
user@host# set ?
```

For more information about configuring local properties for SRC components, see Configuring Basic Local Properties.

3. (Optional) Verify your configuration.

```
[edit slot 0 ims initial]
user@host# show
```

Configuring Directory Connection Properties for IMS

Use the following configuration statements to configure directory connection properties for IMS:

```
slot number ims 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 ims initial directory-connection
```

2. Specify the properties for IMS.

```
[edit slot 0 ims 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 ims 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 IMS

Use the following configuration statements to configure directory eventing properties for IMS:

```
slot number ims 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 ims initial eventing
```

2. Specify the initial directory eventing properties for IMS.

```
[edit slot 0 ims 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 ims initial directory-eventing]
user@host# show
eventing;
polling-interval 30;
```

Configuring the Local Diameter Peer

Use the following configuration statements to configure the local Diameter peer:

```
slot number ims aracf-rq {
  protocol protocol;
  port port;
  address address;
  origin-host origin-host;
  origin-realm origin-realm;
}
```

To configure the local Diameter peer:

1. From configuration mode, access the configuration statement that configures the Diameter peer.

```
user@host# edit slot 0 ims aracf-rq
```

2. (Optional) Specify the protocol used for the transport layer.

```
[edit slot 0 ims aracf-rq]
user@host# set protocol protocol
```

3. (Optional) Specify the port used for incoming connections.

```
[edit slot 0 ims aracf-rq]
user@host# set port port
```

4. (Optional) Specify the IP address of the local peer.

```
[edit slot 0 ims aracf-rq]
user@host# set address address
```

5. (Optional) Specify the Diameter identifier for the local endpoint that is the originator of the Diameter message.

```
[edit slot 0 ims aracf-rq]
user@host# set origin-host origin-host
```

6. (Optional) Specify the Diameter identifier for the realm of the local endpoint that is the originator of the Diameter message.

```
[edit slot 0 ims aracf-rq]
user@host# set origin-realm origin-realm
```

7. (Optional) Verify your configuration.

```
[edit slot 0 ims aracf-rq]
user@host# show
protocol tcp;
port 3868;
address 127.0.0.1;
origin-host testserver;
origin-realm testrealm;
peer 1 {
  address 127.0.0.1;
  origin-host testclient;
}
```

Configuring the Remote Diameter Peer

Use the following configuration statements to configure the remote Diameter peer:

```
slot number ims aracf-rq peer primary-spdf {
  address address;
  origin-host origin-host;
}
```

To configure the remote Diameter peer:

1. From configuration mode, access the configuration statement that configures the Diameter peer. In this sample procedure, the remote SPDF peer called primary-spdf is configured.

```
user@host# edit slot 0 ims aracf-rq peer primary-spdf
```

2. (Optional) Specify the IP address of the remote peer.

```
[edit slot 0 ims aracf-rq peer primary-spdf]
user@host# set address address
```

3. (Optional) Specify the Diameter identifier for the remote endpoint that is the originator of the Diameter message.

```
[edit slot 0 ims aracf-rq peer primary-spdf]
user@host# set origin-host origin-host
```

4. (Optional) Verify your configuration.

```
[edit slot 0 ims aracf-rq peer primary-spdf]
user@host# show
address 127.0.0.1;
origin-host testclient;
```

Configuring Logging Destinations to Store Messages in a File

Use the following configuration statements to configure file logging for IMS:

```
slot number ims logger name ...
slot number ims logger name file {
  filter filter;
  filename filename;
  rollover-filename rollover-filename;
  maximum-file-size maximum-file-size;
}
```

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 log1 is configured.

```
user@host# edit slot 0 ims logger log1 file
```

2. Specify the properties for the logging destination.

```
[edit slot 0 ims logger log1 file]
user@host# set ?
```

For more information about configuring properties for the logging destination, see Overview of Logging for SRC Components.

3. (Optional) Verify your configuration.

```
[edit slot 0 ims logger log1 file]
user@host# show
filter /info-;
filename var/log/ims-a-racf-rq-info.log;
rollover-filename var/log/ims-a-racf-rq-info.alt;
maximum-file-size 2000000000;
```

Configuring Logging Destinations to Send Messages to the System Logging Facility

Use the following configuration statements to configure system logging for IMS:

```
slot number ims logger name ...
slot number ims logger name syslog {
    filter filter;
    host host;
    facility facility;
    format format;
}
```

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 log2 is configured.

```
user@host# edit slot 0 ims logger log2 syslog
```

2. Specify the properties for the logging destination.

```
[edit slot 0 ims logger log2 syslog]
user@host# set ?
```

For more information about configuring properties for the logging destination, see Overview of Logging for SRC Components.

3. (Optional) Verify your configuration.

```
[edit slot 0 ims logger log2 syslog]
user@host# show
```

Configuring the Subscriber Type

Use the following configuration statements to configure the subscriber type:

```
shared ims aracf-rq configuration subscriber-type-configuration name
shared ims aracf-rq configuration subscriber-type-configuration name {
    nic-proxy nic-proxy;
    subscriber-id-type subscriber-id-type;
}
```

To configure the subscriber type:

1. From configuration mode, access the configuration statement that configures the subscriber type. In this sample procedure, the subscriber type called ip is configured.

```
user@host# edit shared ims aracf-rq configuration subscriber-type-configuration ip
```

2. Specify the namespace that defines the properties for the NIC proxy operations for the specified subscriber ID type. Each subscriber type must use a different NIC proxy. All NIC proxies for IMS are stored in the `/nicProxies` directory. In this sample procedure, the namespace for the NIC proxy called `ip` is configured.

```
[edit shared ims aracf-rq configuration subscriber-type-configuration ip]
user@host# set nic-proxy /nicProxies/ip
```

3. (Optional) Specify the type of information used to identify the subscriber. In this sample procedure, the subscriber ID type is specified as the subscriber IP address.

```
[edit shared ims aracf-rq configuration subscriber-type-configuration ip]
user@host# set subscriber-id-type address
```

4. (Optional) Verify your configuration.

```
[edit shared ims aracf-rq configuration subscriber-type-configuration ip]

user@host# show
subscriber-id-type SIT_ADDRESS;
nic-proxy-namespace /nicProxies/ip;
```

Configuring a NIC Proxy for IMS

To configure the NIC proxy, perform these tasks:

1. Configuring Resolution Information for a NIC Proxy on page 163
2. Changing the Configuration for the NIC Proxy Cache on page 165
3. Configuring a NIC Proxy for NIC Replication on page 166
4. Configuring NIC Test Data on page 168

Configuring Resolution Information for a NIC Proxy

You create a NIC proxy for each subscriber type to be configured. Subscriber types that have different subscriber ID types can use the same 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 [Configuring the NIC \(SRC CLI\); Overview of NIC Proxy Configuration](#); and [.](#)

Use the following configuration statements to configure the NIC proxy:

```
shared ims aracf-rq configuration nic-proxy-configuration name
shared ims aracf-rq configuration 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 configures the NIC proxy configuration. In this sample procedure, the NIC proxy called ip is configured.

```

user@host# edit shared ims aracf-rq configuration nic-proxy-configuration ip
resolution

```

2. Specify the NIC resolver that this NIC proxy uses. This resolver must be the same as one that is configured on the NIC host.

```

[edit shared ims aracf-rq configuration nic-proxy-configuration ip resolution]
user@host# set resolver-name resolver-name

```

3. Specify the NIC data type that the key provides for the NIC resolution.

```

[edit shared ims aracf-rq configuration nic-proxy-configuration ip resolution]
user@host# set key-type key-type

```

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 ims aracf-rq 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 ims aracf-rq configuration nic-proxy-configuration ip resolution]
user@host# set value-type value-type

```

5. (Optional) If the key can have more than one value, specify that the key can have multiple corresponding values.

```

[edit shared ims aracf-rq 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 ims aracf-rq configuration nic-proxy-configuration ip resolution]
user@host# set constraints constraints

```

7. (Optional) Verify your configuration.

```
[edit shared ims aracf-rq configuration nic-proxy-configuration ip
resolution]
user@host# show
resolver-name /realms/ip/A1;
key-type Ip;
value-type SaeId;
```

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 change values for the NIC proxy cache:

```
shared ims aracf-rq configuration 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. In this sample procedure, the NIC proxy called ip is configured.

```
user@host# edit shared ims aracf-rq configuration nic-proxy-configuration ip
cache
```

2. (Optional) Specify the maximum number of keys for which the NIC proxy retains data.

```
[edit shared ims aracf-rq 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 ims aracf-rq configuration nic-proxy-configuration ip cache]
user@host# set cache-cleanup-interval cache-cleanup-interval
```

4. (Optional) Specify how long an entry remains in the cache.

```
[edit shared ims aracf-rq configuration nic-proxy-configuration ip cache]
user@host# set cache-entry-age cache-entry-age
```

5. (Optional) Verify your configuration.

```
[edit shared ims aracf-rq configuration nic-proxy-configuration ip cache]

user@host# show
cache-size 10000;
cache-cleanup-interval 15;
```

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:

```
shared ims aracf-rq configuration nic-proxy-configuration name nic-host-selection {
  groups groups ;
  selection-criteria (roundRobin | randomPick | priorityList);
}
shared ims aracf-rq configuration 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. In this sample procedure, the NIC proxy called ip is configured.

```
user@host# edit shared ims aracf-rq configuration nic-proxy-configuration ip
nic-host-selection
```

2. (Optional) Specify the list of groups of NIC hosts that the NIC proxy can contact for resolution requests.

```
[edit shared ims aracf-rq configuration nic-proxy-configuration ip nic-host-selection]
user@host# set groups groups
```

3. (Optional) 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 ims aracf-rq 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. (Optional) Verify your configuration.

```
[edit shared ims aracf-rq configuration nic-proxy-configuration ip
nic-host-selection]
user@host# show
groups ;
selection-criteria round-;
```

5. Access the configuration statement that specifies the NIC proxy configuration for blacklisting—the process of handling nonresponsive NIC hosts.

```
[edit shared ims aracf-rq configuration nic-proxy-configuration ip nic-host-selection]
user@host# edit blacklisting
[edit shared ims aracf-rq configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
```

6. (Optional) 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 ims aracf-rq configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
user@host# set try-next-system-on-error
```

7. (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 ims aracf-rq configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
user@host# set number-of-retries-before-blacklisting
number-of-retries-before-blacklisting
```

8. (Optional) Change the interval at which the NIC proxy attempts to connect to an unavailable NIC host. The default is 15 seconds.

```
[edit shared ims aracf-rq configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
user@host# set blacklist-retry-interval blacklist-retry-interval
```

9. (Optional) Verify your configuration.

```
[edit shared ims aracf-rq configuration nic-proxy-configuration ip
nic-host-selection blacklisting]
user@host# show
try-next-system-on-error;
number-of-retries-before-blacklisting 3;
blacklist-retry-interval 15;
```

Configuring NIC Test Data

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 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.

Use the following configuration statements to configure a NIC proxy stub from the [edit] hierarchy level.

```
shared ims aracf-rq configuration nic-proxy-configuration name test-nic-bindings {
  use-test-bindings;
}
shared ims aracf-rq configuration nic-proxy-configuration name test-nic-bindings
  key-values name {
    value ;
  }
```

To use the NIC proxy stub for IMS:

1. In configuration mode, navigate to the NIC proxy configuration and specify the data type of the key you want to map to a value. In this sample procedure, the key `ip` is specified for the NIC proxy called `ip`.

```
[edit shared ims aracf-rq configuration nic-proxy-configuration ip]
user@host# set resolution key-type ip
```

2. Enable a NIC proxy stub for a resolution.

```
[edit shared ims aracf-rq configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings use-test-bindings
```

3. Specify the values of the keys for testing. These statements are available at the Advanced CLI editing level.

```
[edit shared ims aracf-rq 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 ims aracf-rq configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings key-values jane@virneo.com 192.0.2.30
```

Managing IMS

After you have configured IMS, you can perform these tasks:

- Starting the IMS Process on page 169
- Restarting the IMS Process on page 169
- Stopping the IMS Process on page 169
- Displaying IMS Status on page 170

Starting the IMS Process

To start the IMS process:

```
user@host> enable component ims
```

The system responds with a start message. If IMS is already running, the system responds with a warning message.

- Related Topics**
- Restarting the IMS Process
 - Stopping the IMS Process

Restarting the IMS Process

You must restart the IMS process after you commit a configuration change.

To restart IMS:

```
user@host> restart component ims
```

The system responds with a start message. If IMS is already running, the system responds with a shutdown message and then a start message.

- Related Topics**
- Starting the IMS Process
 - Stopping the IMS Process

Stopping the IMS Process

To stop the IMS process:

```
user@host> disable component ims
```

The system responds with a shutdown message. If IMS is not running when you issue the command, the system responds with the command prompt.

- Related Topics**
- Starting the IMS Process
 - Restarting the IMS Process
 - Displaying IMS Status

Displaying IMS Status

Purpose Display IMS status.

Action user@host> **show component**

The system responds with a status message.

- Related Topics**
- Configuring the IMS Software
 - Monitoring IMS with the SRC CLI
 - Stopping the IMS Process

Monitoring IMS with the SRC CLI

Monitoring tasks are:

1. Viewing Server Process Information on page 170
2. Viewing Statistics for the Rq Interface on page 170

Viewing Server Process Information

Purpose View information about the IMS server process.

Action user@host> **show ims statistics aracf rq process**
Rq Server Process
 Rq server up time (seconds) 692942
 Rq server up since 2007-03-13T15:30:48EDT
 Rq server threads 93
 Heap used (bytes) 16383752 (8%)
 Heap limit (bytes) 200000000

Viewing Statistics for the Rq Interface

Purpose Monitor the current state of the A-RACF Rq interface.

Action user@host> **show ims statistics aracf rq**
ims aracf rq Statistics
Rq Server Process
 Rq server up time (seconds) 692920
 Rq server up since 2007-03-13T15:30:48EDT

Rq server threads	93
Heap used (bytes)	16332120 (8%)
Heap limit (bytes)	200000000

Monitoring IMS with the C-Web Interface

You can monitor statistics for the server process and the A-RACF Rq interface with the C-Web interface by:

1. Viewing Statistics for the Server Process on page 171
2. Viewing Statistics for the A-RACF Rq Interface on page 171

Viewing Statistics for the Server Process

Purpose View statistics for the server process.

Action Select **IMS** from the side pane, click **Statistics**, click **A-RACF**, click **Rq**, and then click **Process**.

The Process pane displays statistics for the server process.

The screenshot shows the Juniper C-Web interface. On the left is a sidebar with a 'Monitor' section containing links to ACP, CLI, Component, Date, Disk, **IMS** (highlighted), Interfaces..., JPS, NIC, NTP, Redirect Server, Route..., SAE, Security, and System. The main content area is titled 'IMS' and 'Process'. It displays a table of statistics for the 'Rq Server Process':

Rq Server Process	
Rq server up time (seconds)	8664
Rq server up since	2007-04-12T14:40:00EDT
Rq server threads	93
Heap used (bytes)	5026424 (3%)
Heap limit (bytes)	200000000

At the top right of the interface, it says 'Logged in as: sleswayball' with links for 'About', 'Refresh', and 'Logout'. Below the table, there is a breadcrumb trail: 'IMS > Statistics > A-RACF > Rq > Process'. The footer contains copyright information for Juniper Networks, Inc. and the Juniper logo.

Viewing Statistics for the A-RACF Rq Interface

Purpose View statistics for the A-RACF Rq interface.

Action Select **IMS** from the side pane, click **Statistics**, click **A-RACF**, and then click **Rq**.

The Rq pane displays statistics for the A-RACF Rq interface.

The screenshot shows the Juniper SRC 3.0.x Monitor interface. The top navigation bar includes 'Monitor', 'Logged in as: sleswayball', 'About', 'Refresh', and 'Logout'. The left sidebar lists various components: ACP, CLI, Component, Date, Disk, IMS (selected), Interfaces..., JPS, NIC, NTP, Redirect Server, Route..., SAE, Security, and System. The main content area displays 'IMS' statistics for 'Rq'. The title is 'ims aracf rq Statistics'. Below this, a table titled 'Rq Server Process' shows the following data:

Rq server up time (seconds)	9373
Rq server up since	2007-04-12T14:40:00EDT
Rq server threads	93
Heap used (bytes)	6013200 (3%)
Heap limit (bytes)	200000000

The footer contains copyright information: 'Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy.' and the Juniper logo with the tagline 'Juniper your Net.'.

Example: Configuring JUNOS Policies for IMS with the SRC CLI

For IMS environments, you can configure JUNOS policies. When you configure classify-traffic conditions, you can set up the software so that the SAE expands into multiple classifiers before it installs the policy on the router. If you enter a list of values in the source and destination network (IP address, mask, and IP operation) or port fields (for port-related protocols), the software creates a classifier for each possible combination of address and port. Note that the software does not expand classifiers for values that are entered as a range.

Enabling Expansion of JUNOS Classify-Traffic Conditions

To enable the expansion of JUNOS classify-traffic conditions:

1. From configuration mode, access the configuration statement that configures policy management properties on the SAE.

```
user@host# edit shared sae configuration policy-management-configuration
```

2. Specify whether or not the SAE expands the JUNOS classify-traffic conditions into multiple classifiers before it installs the policy on the router.

```
[edit shared sae configuration policy-management-configuration]
user@host# set enable-junos-classifier-expansion
```

Related Topics ■ For more information about expanded classifiers, see Policy Information Model

Part 5

Index

- Index on page 175

Index

A

- address pools
 - assigned IP subscribers
 - configuring.....94
- address pools. *See* IP address pools
- application manager
 - role, in PCMM environment.....39
- assigned IP subscribers
 - PCMM network.....49, 68
 - address pools.....94
 - IP address pools.....49
 - setting timeouts.....25
 - voice over IP.....24

C

- cable modem termination system. *See* CMTS devices
- classify-traffic condition
 - expanded classifiers
 - configuring.....172
- client type 1, PCMM.....41
- client type 2, PCMM.....42
- CMTS devices
 - adding objects to directory
 - SRC CLI.....67
 - adding virtual router objects to directory
 - SRC CLI.....68
 - configuration statements.....67, 68
 - role.....39
- CMTS locator
 - monitoring
 - C-Web interface.....109
 - SRC CLI.....103
- CoA script services, configuring.....115
- conventions
 - notice icons.....xix
 - text.....xix
- custom RADIUS authentication plug-ins.....16
- customer support.....xxiii
 - contacting JTAC.....xxiii

D

- Data over Cable Service Interface Specifications. *See* DOCSIS protocol
- Diameter server
 - clients, viewing
 - SRC CLI.....142
 - message flows, viewing
 - SRC CLI.....142
 - message handler, viewing
 - SRC CLI.....142
 - monitoring
 - SRC CLI.....141
 - peers, viewing
 - SRC CLI.....142
 - server process, viewing
 - SRC CLI.....142
 - server requests, viewing
 - SRC CLI.....142
 - state, viewing
 - SRC CLI.....142
 - statistics, viewing
 - SRC CLI.....142
- DOCSIS protocol.....40
- documentation set
 - comments on.....xxiii
- domains
 - IP service edge.....44
 - IP subscriber edge.....44
 - radio frequency.....44
- dynamic RADIUS authorization requests
 - RADIUS packets, defining.....35, 120

E

- end-to-end services.....44
- event notification, PCMM network
 - configuration statements.....60
 - description.....50
 - properties, configuring
 - SRC CLI.....60
- expanded classifiers
 - configuring.....172

F

flexible RADIUS authentication plug-ins	
configuring.....	17

I

IP address pools	
assigned IP subscribers.....	49
assigned IP subscribers, configuring	
SRC CLI.....	68
local address pools, configuring	
SRC CLI.....	68
static pools, configuring	
SRC CLI.....	68

J

JPS (Juniper Policy Server)	
application manager-to-policy server interface,	
configuring.....	83
application manager-to-policy server interface,	
monitoring	
C-Web interface.....	106, 107
SRC CLI.....	102
architecture.....	73
CMTS devices, monitoring	
C-Web interface.....	108
CMTS locator, monitoring	
C-Web interface.....	109
SRC CLI.....	103
JPS state, monitoring.....	102
logging, configuring.....	82
logging, modifying.....	82
message flows, monitoring	
C-Web interface.....	110
SRC CLI.....	103
message handler, monitoring	
C-Web interface.....	109
SRC CLI.....	103
monitoring	
C-Web interface.....	105
SRC CLI.....	99, 101
operational status.....	99
overview.....	73
policy server-to-CMTS interface, configuring.....	89
policy server-to-CMTS interface, monitoring	
C-Web interface.....	107, 108
SRC CLI.....	102
policy server-to-RKS interface, configuring.....	85
policy server-to-RKS interface, monitoring	
C-Web interface.....	111
SRC CLI.....	102
server process, monitoring	
C-Web interface.....	105
SRC CLI.....	101

starting	
SRC CLI.....	99
stopping	
SRC CLI.....	99
subscriber address mappings, configuring.....	92
subscriber configuration, modifying.....	92

Juniper Policy Server. *See* JPS

L

login process	
assigned IP subscribers, PCMM.....	49

M

manuals	
comments on.....	xxiii

N

NIC (network information collector)	
IP address pools, configuring	
SRC CLI.....	68
NIC (network information collector)	
testing	
test data.....	168
NIC proxies	
cache, configuring	
SRC CLI.....	165
configuration prerequisites.....	163
NIC replication, configuring	
SRC CLI.....	165
resolution information, configuring	
SRC CLI.....	163
notice icons.....	xix

P

packet mirroring, configuring.....	28
PCMM (PacketCable Multimedia)	
application manager, role.....	39
client type 1.....	41
client type 2.....	42
CMTS device, role.....	39
configuring SAE	
SRC CLI.....	55
creating sessions.....	49
description.....	39
end-to-end QoS architecture.....	44
end-to-end services.....	44
integrating SRC software.....	39
IP service edge domain.....	44
IP subscriber edge domain.....	44
logging in subscribers	
assigned IP method.....	49
overview.....	49

overview.....	39
policy server, role.....	39
provisioning end-to-end services.....	46
record-keeping server.....	39
RF domain.....	39
SAE.....	49
SAE communities.....	52
session store.....	53
single-phase resource reservation model.....	41
SRC software in	
description.....	43
traffic profiles.....	43
video-on-demand example.....	47
videoconferencing example.....	46
PCMM device driver	
configuration statements.....	56
configuring	
SRC CLI.....	56
PCMM record-keeping server plug-in	
configuration statements.....	62
configuring	
SRC CLI.....	62
description.....	53
plug-ins	
PCMM record-keeping server plug-in.....	53
policy servers	
adding application manager groups	
SRC CLI.....	94
adding objects to directory	
SRC CLI.....	96
role, in PCMM architecture.....	39
specifying application managers	
SRC CLI.....	94
specifying SAE communities	
SRC CLI.....	94
priorityList.....	167

Q

QoS (quality of service)	
PCMM environments	
end-to-end QoS architecture.....	44
extending to service edge domain.....	46
extending to subscriber edge domain.....	45
searching for policies in directory.....	14
QoS profile-tracking plug-in	
description.....	3
QoS profiles, JUNOS routers	
how tracking works.....	3
managing dynamically.....	3
updating directory, using	
qosProfilePublish.....	12
quality of service. <i>See</i> QoS	

R

RADIUS	
vendor-specific attributes for wireless ISP	
roaming.....	17
randomPick.....	167
record-keeping server. <i>See</i> RKS	
RKS (record-keeping server)	
peers, configuration statements.....	61
peers, configuring in plug-ins	
SRC CLI.....	61
plug-in.....	53
plug-in, configuration statements.....	64
plug-in, configuring	
SRC CLI.....	62
role in PCMM environment.....	39
roaming wireless environment.....	15
roundRobin.....	167

S

SAE (service activation engine)	
configuring as an application manager	
SRC CLI.....	94
PCMM environment.....	49
redundancy. <i>See</i> SAE communities	
SAE (service activation engine), configuring	
community manager	
SRC CLI.....	59
event notification API properties	
SRC CLI.....	60
PCMM device driver	
SRC CLI.....	56
SAE communities	
configuration overview	
SRC CLI.....	58
configuration statements.....	59
configuring manager	
SRC CLI.....	59
defining members	
SRC CLI.....	68
description.....	52
service flows.....	41
services	
voice over IP (VoIP).....	23
session store	
in PCMM environment.....	53
single phase resource reservation model, PCMM.....	41
subscriber	
wireless environment.....	15
support, technical <i>See</i> technical support	

T

technical support	
contacting JTAC.....	xxiii
text conventions defined.....	xix

traffic policies, PCMM.....43

W

wireless environment.....15