



SRC-PE Software

Subscribers and Subscriptions Guide

Release 3.1.x

Juniper Networks, Inc.

1194 North Mathilda Avenue
Sunnyvale, California 94089
USA

408-745-2000

www.juniper.net

Part Number: 530-028670-01, Revision 1

Juniper Networks, the Juniper Networks logo, JUNOS, NetScreen, ScreenOS, and Steel-Belted Radius are registered trademarks of Juniper Networks, Inc. in the United States and other countries. JUNOS is a trademark of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

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

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

SRC-PE Software Subscribers and Subscriptions Guide

Release 3.1.x

Copyright © 2009, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Writing: Linda Creed, Justine Kangas, Betty Lew, Helen Shaw

Editing: Fran Mues

Illustration: Nathaniel Woodward

Cover Design: Edmonds Design

Revision History

13 February 2009—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 (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

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

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

a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

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

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

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

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

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

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

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

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

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

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

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

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

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous

agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

About This Guide

xix

Part 1

Managing Subscribers and Subscriptions

Chapter 1	Overview of Subscribers and Subscriptions on a C-series Controller	3
Chapter 2	Subscriber Logins and Service Activation	9
Chapter 3	Configuring Subscriber-Related Properties on the SAE (SRC CLI)	35
Chapter 4	Classifying Interfaces and Subscribers (SRC CLI)	43
Chapter 5	Overview of Plug-Ins Included with the SAE	81
Chapter 6	Configuring Internal, External, and Synchronization Plug-Ins (SRC CLI)	89
Chapter 7	Configuring Accounting and Authentication Plug-Ins (SRC CLI)	93
Chapter 8	Configuring Subscribers and Subscriptions (SRC CLI)	135

Part 2

Redirecting Subscriber Traffic Through Redirect Server

Chapter 9	Redirecting Subscriber Traffic	161
Chapter 10	Configuring Traffic Redirection (SRC CLI)	165

Part 3

Integrating JUNOS VPNs in to an SRC Configuration

Chapter 11	Adding VPNs from JUNOS Routing Platforms (SRC CLI)	183
------------	--	-----

Part 4

Index

Index	191
-------	-----

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

Managing Subscribers and Subscriptions

Chapter 1

Overview of Subscribers and Subscriptions on a C-series Controller 3

Overview of Subscribers	3
Overview of Subscriptions	4
Enterprise Subscriber and Subscription Hierarchy	4
Enterprise Subscription Hierarchy	6
Overview of Managers	6
Read Privileges	6
Management Privileges	6
Managers That Control All Retailers	7

Chapter 2

Subscriber Logins and Service Activation 9

Login Events and Processes for the SRC Software	9
Overview of Login Events and Processes	9
Login Events	10
Summary of the Login Process	11
Residential Subscriber Login and Processes	11
PPP Subscriber Login and Service Activation	13
Web Login for PPP Subscribers	13
PPP Login Interactions	14
PPP Logout Interactions	15
DHCP Subscriber Login and Service Activation	17
Interface Startup	17
Initial Login	17
Initial DHCP Login Interactions	18

DHCP Login to Subscriber Account Interactions	19
Persistent DHCP Subscriber Login Interactions	21
DHCP Subscriber Logout Interactions	22
Static IP Subscribers	23
Single PC, IP Address Known	23
Subscriber IP Address Not Known	24
Enterprise Subscriber Login Process	26
Interface Startup	26
Subscriptions and Activations	27
Subscription Activation Interactions	29
Subscription Deactivation Interactions	31
Automatic Activation at Login	32
Enterprise-Specific Remote Session Activation	33

Chapter 3**Configuring Subscriber-Related Properties on the SAE (SRC CLI) 35**

Configuring the Length of Time MAC Addresses Remain in SAE Cache	35
Identifying a Profile for Unauthenticated Subscribers	36
Configuring Interim Accounting for Services and Subscribers	37
Avoiding Overcharges for Sessions That Time Out	38
Allowing Multiple Logins from the Same IP Address	39
Authenticating Registered Username/Password Pairs	40
Configuring Timers for Session Reactivation	40

Chapter 4**Classifying Interfaces and Subscribers (SRC CLI) 43**

Overview of Classification Scripts	43
How Classification Scripts Work	44
Interface Classification Scripts	44
Subscriber Classification Scripts	45
DHCP Classification Scripts	45
Sharing Information Among Classification Scripts	45
Overview of Configuring Classification Scripts	46
Subscriber Classifiers	46
DHCP Classifiers	46
Interface Classifiers	46
Classification Targets	47
Target Expressions	47
Classification Conditions	47
Glob Matching	48
Regular Expression Matching	48
Classifying Interfaces (SRC CLI)	49
Interface Classification Conditions	52
Example: Managing Interfaces for Premium and Basic PPP and DHCP	
Subscribers	54
Example: Managing Specific Interfaces	55
Example: Managing Interfaces by Using the Interface Description	55
Classifying Subscribers (SRC CLI)	56
Subscriber Classification Conditions	59

Sending DHCP Options to the JUNOSe Router	63
Subscriber Classification Targets	64
Example: Subscriber Classification Scripts for Static IP Subscriber	65
Example: Subscriber Classification Scripts Using a Subscriber Group	66
Example: Subscriber Classification Scripts for Enterprise Subscribers	66
Matching on the Interface Name	66
Matching on the Interface Alias	67
Example: Creating Router Interface Subscriber Session	67
Example: Activating Services for a Group of Subscriber Sessions	68
Classifying DHCP Subscribers (SRC CLI)	68
DHCP Classification Conditions	70
Syntax for DHCP Classification Targets	72
Selecting DHCP Parameters	73
DHCP Options Supported on the SAE	74
Creating DHCP Profiles (SRC CLI)	77

Chapter 5 Overview of Plug-Ins Included with the SAE 81

How Internal Plug-Ins Work	81
Plug-In Pool	81
Event Publishers	82
Types of Internal Plug-Ins	82
Authorization Plug-Ins	82
Tracking Plug-Ins	83
Customizing RADIUS Packets with Plug-Ins	84
Assigning DHCP Addresses to Subscribers	84
Creating and Tracking Subscriber Sessions	86
Activating and Tracking Service Sessions	87

Chapter 6 Configuring Internal, External, and Synchronization Plug-Ins (SRC CLI) 89

Configuring Internal Plug-Ins	89
Configuring the SAE for External Plug-Ins	90
Configuring the State Synchronization Plug-In Interface	91

Chapter 7 Configuring Accounting and Authentication Plug-Ins (SRC CLI) 93

Creating RADIUS Peers	93
Types of Tracking Plug-Ins	95
Configuring Tracking Plug-Ins	96
Configuring Flat File Accounting Plug-Ins	96
Configuring Headers for Flat File Accounting Plug-Ins	98
Configuring Basic RADIUS Accounting Plug-Ins	99

Configuring Flexible RADIUS Accounting Plug-Ins	101
Configuring Custom RADIUS Accounting-Plug-Ins	104
Types of Authentication Plug-Ins	106
Configuring Authentication Plug-Ins	107
Limiting Subscribers on Router Interfaces	108
Configuring Basic RADIUS Authentication Plug-Ins	108
Configuring Flexible RADIUS Authentication Plug-Ins	110
Configuring Custom RADIUS Authentication Plug-Ins	112
Configuring LDAP Authentication Plug-Ins	115
Configuring UDP Ports for RADIUS Plug-Ins	117
Defining RADIUS Packets for Flexible RADIUS Plug-Ins	118
Overview of Flexible RADIUS Plug-Ins	118
Using Default RADIUS Templates	119
Naming RADIUS Attribute Instances	119
Defining RADIUS Attributes	120
Standard RADIUS Attributes	120
Juniper Networks VSAs	120
Defining the Values of RADIUS Attributes	121
Configuring a RADIUS Packet Template	125
Using Flexible RADIUS Packet Definitions	127
Setting Values in Authentication Response Packets	129
Selecting IP Address Pools Using DHCP Response Packets	129
Configuring Event Publishers	130
Special Types of Event Publishers	130
Configuring Service-Specific Event Publishers	130
Configuring Retailer-Specific Event Publishers	130
Configuring Virtual Router-Specific Event Publishers	131
Configuring Global and Default Retailer Event Publishers	131

Chapter 8

Configuring Subscribers and Subscriptions (SRC CLI) 135

Overview of Configuring Subscribers and Subscriptions	135
Specifying the Activation Order for Subscriptions	135
Inheritance of Properties and Subscriptions	136
Enabling the Subscriber and Subscription Configuration on the SRC CLI	136
Adding Subscribers (SRC CLI)	137
Adding Retailers (SRC CLI)	137
Configuring Administrative Information for Retailers (SRC CLI)	139
Adding Subscriber Folders (SRC CLI)	140
Adding Residential Subscribers (SRC CLI)	141
Configuring Administrative Information for Residential Subscribers (SRC CLI)	144
Adding Enterprises (SRC CLI)	145
Configuring Administrative Information for Enterprise Subscribers (SRC CLI)	147
Adding Sites (SRC CLI)	148
Adding Devices as Subscribers (SRC CLI)	149
Adding Managers (SRC CLI)	151
Configuring Subscriptions (SRC CLI)	153
Configuring Accesses (SRC CLI)	155

Part 2	Redirecting Subscriber Traffic Through Redirect Server	
Chapter 9	Redirecting Subscriber Traffic	161
	Overview of Traffic Redirection	161
	Proxy Request Management	161
	HTTP Proxy and DNS	162
	Protection Against Denial-of-Service Attacks	163
	Redirect Server Redundancy	163
Chapter 10	Configuring Traffic Redirection (SRC CLI)	165
	Configuration Statements for the Redirect Server (SRC CLI)	165
	Before You Configure the Redirect Server on a C-Series Controller	166
	Configuring the Redirect Server (SRC CLI)	167
	Configuring General Properties for the Redirect Server (SRC CLI)	168
	Configuring a Connection Between the Redirect Server and the Directory (SRC CLI)	169
	Defining Traffic to Transmit to the Redirect Server (SRC CLI)	170
	Changing the Number of Requests That the Redirect Server Accepts (SRC CLI)	171
	Specifying Extensions for Files That the Redirect Server Accepts (SRC CLI)	172
	Verifying Configuration for the Redirect Server (SRC CLI)	173
	Enabling the Redirect Server	173
	Configuring the DNS Server for the Redirect Server (SRC CLI)	174
	Configuring the Redirect Server to Support HTTP Proxies (SRC CLI)	175
	Before You Configure Redundancy for a Redirect Server	176
	Configuring a Redundant Redirect Server (SRC CLI)	176
	Configuring Logging for the Redirect Server	178
	Changing the Configuration for the Redirect Server	178
	Assessing Load for Redirect Server (C-Web Interface)	178
Part 3	Integrating JUNOS VPNs in to an SRC Configuration	
Chapter 11	Adding VPNs from JUNOS Routing Platforms (SRC CLI)	183
	Before You Add a JUNOS VPN to the SRC Configuration	183
	Configuring VPNs to Integrate into an SRC Network	184
	Configuration Statements for Adding VPNs and Extranet Clients	184
	Adding VPNs for Retailers and Enterprises	185
	Verifying and Updating Configuration of Extranets for VPNs	186
	Locating and Removing Inactive Subscriptions to a VPN	187

Part 4

Index

Index191

List of Figures

Figure 1: Enterprise Hierarchy	5
Figure 2: Components Involved in Subscription Activation	12
Figure 3: PPP Login Interactions	14
Figure 4: PPP Logout	16
Figure 5: DHCP Interface Startup	17
Figure 6: DHCP Subscriber Initial Login	18
Figure 7: DHCP Subscriber Login	20
Figure 8: Persistent DHCP Subscriber Login	21
Figure 9: DHCP Subscriber Logout	22
Figure 10: Static IP Subscriber Login	24
Figure 11: Subscriber IP Address Not Known	26
Figure 12: Enterprise Subscriber Session Activation	27
Figure 13: Service Activation Page	28
Figure 14: Subscription Activation Page	29
Figure 15: Subscription Activation	30
Figure 16: Subscription Deactivation	31
Figure 17: Remote Session Activation Sequence	33
Figure 18: DHCP Address Assignment	84
Figure 19: Creating and Tracking Subscriber Sessions	86
Figure 20: Activating and Tracking Service Sessions	87
Figure 21: Failover of a Redirect Server	164

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: Types of Subscribers	3
Table 5: Privilege Levels and Associated Tasks	7
Table 6: Login Events	10
Table 7: DHCP Options in UserClassificationContext Field	63
Table 8: DHCP Options Supported on the SAE	74
Table 9: Tracking Plug-Ins	95
Table 10: Authentication Plug-Ins	107
Table 11: RADIUS Attribute Instance Names	119
Table 12: Standard Values for RADIUS Attributes	121

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

The most current SRC documentation is available at:

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

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

Obtaining Documentation

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

To download complete sets of technical documentation to create your own documentation CD-ROMs or DVD-ROMs, see the CD-ROM and DVD-ROM Documentation page at

<http://www.juniper.net/techpubs/resources/cdrom.html>

Copies of the Management Information Bases (MIBs) are available 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 <https://www.juniper.net/cgi-bin/docbugreport/>. 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 Management tool: <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 Management tool in the CSC at <http://www.juniper.net/cm/> .
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

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

Part 1

Managing Subscribers and Subscriptions

- Overview of Subscribers and Subscriptions on a C-series Controller on page 3
- Subscriber Logins and Service Activation on page 9
- Configuring Subscriber-Related Properties on the SAE (SRC CLI) on page 35
- Classifying Interfaces and Subscribers (SRC CLI) on page 43
- Overview of Plug-Ins Included with the SAE on page 81
- Configuring Internal, External, and Synchronization Plug-Ins (SRC CLI) on page 89
- Configuring Accounting and Authentication Plug-Ins (SRC CLI) on page 93
- Configuring Subscribers and Subscriptions (SRC CLI) on page 135

Chapter 1

Overview of Subscribers and Subscriptions on a C-series Controller

- Overview of Subscribers on page 3
- Overview of Subscriptions on page 4
- Enterprise Subscriber and Subscription Hierarchy on page 4
- Overview of Managers on page 6

Overview of Subscribers

A subscriber is an object in the directory for which you can configure subscriptions to services. The SRC software distinguishes between types of subscribers, as described in Table 4 on page 3.

Table 4: Types of Subscribers

Subscriber	Description
Retailers	Internet service providers who either manage their own subscribers or outsource the management of subscribers to a service provider who deploys the SRC software. The SRC software uses retailer objects to group subscribers who belong to an administrative domain.
Residential	Individual subscribers or households—multiple subscribers who use one or more computers and share the same connection. In a household, subscribers can share the same service subscription or can have their own individualized service profiles.
Enterprise	An organization, such as a corporation. An enterprise subscriber can contain site subscribers that represent physical locations or groups within the organization. Enterprises and sites contain access subscribers; an access represents a layer 2 connection between a device at a customer's physical location and a router that gives the enterprise subscribers access to the Internet and, in some cases, a virtual private network (VPN).
Sites	One or more locations—physical or virtual—within an enterprise that share service subscriptions and physical access to services and that are each managed as a unique entity. For example, the XYM Corporation might have a site in Boston and a site in Toronto. Each of these sites can have its own set of subscribed services.

Table 4: Types of Subscribers *(continued)*

Subscriber	Description
Device	An SRC-managed device that is used to activate services on nonsubscriber interfaces. It is used primarily to provide integration with applications that use traffic mirroring on JUNOS routing platforms.
Subscriber folders	Objects that group subscribers.

- Related Topics**
- Overview of Subscriptions on page 4
 - Enterprise Subscriber and Subscription Hierarchy on page 4
 - Overview of Configuring Subscribers and Subscriptions on page 135
 - Enabling the Subscriber and Subscription Configuration on the SRC CLI on page 136
 - Adding Subscribers (SRC CLI) on page 137
 - Adding Subscriber Folders (SRC CLI) on page 140

Overview of Subscriptions

A subscription is an object that represents an enrollment to a service. Each subscription provides access to a particular service for that subscriber. A subscriber can have multiple subscriptions to a service.

If the service provider uses the SRC directory to hold all their subscriber data, residential subscribers must subscribe to primary services—such as Broadband Remote Access Server (B-RAS) through Point-to-Point protocol (PPP) or B-RAS through Dynamic Host Configuration Protocol (DHCP)—before subscribing to a service.

Enterprise subscribers must subscribe to an access (that is, a leased line), either directly or in a site or subscriber folder that is subordinate to the enterprise. Without an access subscription, a service session cannot run in the network.

- Related Topics**
- Overview of Subscribers on page 3
 - Enterprise Subscriber and Subscription Hierarchy on page 4
 - Subscriptions and Activations on page 27
 - Overview of Configuring Subscribers and Subscriptions on page 135
 - Configuring Subscriptions (SRC CLI) on page 153

Enterprise Subscriber and Subscription Hierarchy

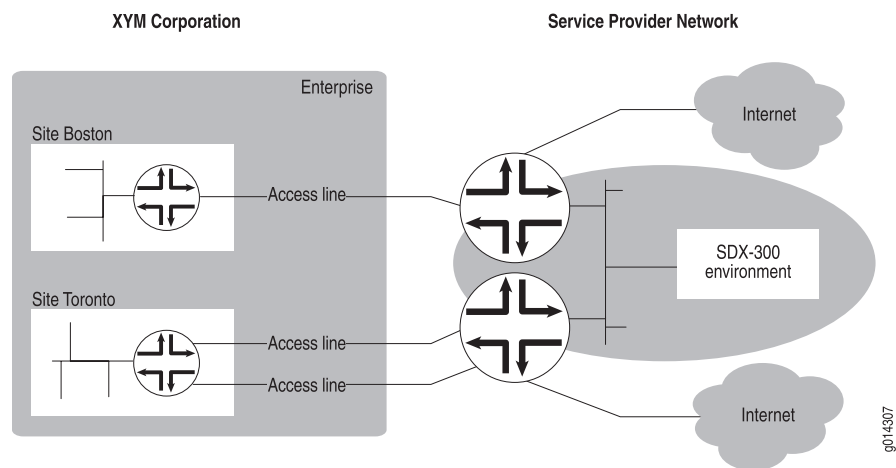
In the enterprise model, a subscriber is an individual physical access line managed through the enterprise service portal over which services are delivered by the service provider. In the enterprise, the SRC software supports the organization of the enterprise in the following hierarchy (Figure 1 on page 5):

- Enterprise—The business itself as a customer of the service provider; for example, the XYM Corporation. An enterprise can have its own set of subscriptions over a physical access line.
- Site—One or more locations, physical or virtual, within the enterprise that share service subscriptions and physical access to services and that are each managed as a unique entity. For example, the XYM Corporation might have a site in Boston and a site in Toronto. Each of these sites can have its own set of subscribed services.
- Access line—A physical access line (usually within a site) from the customer to the service provider's router; the router is configured to access the SRC environment and the Internet and/or the customer's network-based VPN. An access line can have its own set of subscribed services.

Enterprise IT managers can use the enterprise service portal to manage interfaces connecting enterprise sites to the network. These interfaces can be leased-line connections or authenticated PPP and DHCP connections.

Figure 1 on page 5 shows an enterprise hierarchy.

Figure 1: Enterprise Hierarchy



Sites and access lines are subordinate to an enterprise; the enterprise sometimes contains sites and access lines. Access lines are subordinate to a site; the site contains access lines.

In Figure 1 on page 5, the XYM Corporation enterprise contains two subordinate sites, Boston and Toronto. The Boston site contains a single subordinate access line, whereas the Toronto site contains two subordinate access lines. All three access lines connect to a router in the service provider network. An individual access line, for example, might be a T1 line running PPP or a T3 line running Frame Relay.

Enterprise Subscription Hierarchy

The organizational levels of the enterprise receive subscribed services in a hierarchical manner. The availability of a subscription to a higher level affects its availability to a lower level.

- Enterprise—Subscriptions apply to all sites and all access lines across the enterprise.
- Site—Subscriptions apply to all access lines grouped within a site.
- Access line—Subscriptions apply to a given access line that connects the enterprise to the service provider's network.

Related Topics

- Overview of Subscribers on page 3
- Overview of Subscriptions on page 4
- Enterprise Subscriber Login Process on page 26
- Overview of Configuring Subscribers and Subscriptions on page 135
- Enabling the Subscriber and Subscription Configuration on the SRC CLI on page 136

Overview of Managers

In relation to subscribers and subscriptions, a manager is an object that represents an IT manager in an organization. Retailers, subscriber folders, enterprises, sites, and accesses can support one or more managers.

Read Privileges

Managers have privileges to read:

- The objects they control
- Parent subscribers, up to the retailer
- Subscriptions of parent subscribers, up to the retailer
- All objects that represent services, service scopes, policies, and global variables that are defined for the subscriber to which the manager is added

Management Privileges

You can specify one or more management privileges for managers. If you do not specify privileges for a manager, the manager has only read privileges. Table 5 on page 7 shows the privilege levels and the privileges associated with the levels.

Table 5: Privilege Levels and Associated Tasks

Privilege Level	Tasks That Managers with This Privilege Can Perform
Administrator	<ul style="list-style-type: none"> ■ Add, delete and modify managers ■ Add, delete, and modify subscriptions ■ Modify subscribers, including the ability to add, delete, and modify substitutions for subscribers ■ Manually activate and deactivate subscription sessions
Subscription	<ul style="list-style-type: none"> ■ Add, delete, and modify subscriptions ■ Manually activate and deactivate subscription sessions
Substitution	Add, delete, and modify substitutions in subscribers and subscriptions
Activation	<ul style="list-style-type: none"> ■ Configure automatic activation of services ■ Manually activate and deactivate subscription sessions
VPNs	Modify, export, and cancel the export of VPNs

A manager has management privileges for its associated subscriber and for that subscriber's subordinate objects:

- Managers in an enterprise have control over the enterprise and all sites and accesses in the enterprise.
- Managers in a site have control over the site and all accesses it contains. In addition they have read access to the enterprise, subscriber folder, and retailer that are configured above the site.
- Managers in an access have control over only that access.

Managers That Control All Retailers

You can add managers that have control over all retailers and their subordinate enterprises. To do so, configure the manager at the [edit subscribers retailer name manager] hierarchy.

- Related Topics**
- Overview of Subscriptions on page 4
 - Overview of Subscribers on page 3
 - Overview of Configuring Subscribers and Subscriptions on page 135
 - Adding Managers (SRC CLI) on page 151

Chapter 2

Subscriber Logins and Service Activation

- Login Events and Processes for the SRC Software on page 9
- Residential Subscriber Login and Processes on page 11
- PPP Subscriber Login and Service Activation on page 13
- DHCP Subscriber Login and Service Activation on page 17
- Static IP Subscribers on page 23
- Enterprise Subscriber Login Process on page 26
- Subscriptions and Activations on page 27
- Automatic Activation at Login on page 32

Login Events and Processes for the SRC Software

Login interactions between the components differ according to the type of subscriber. Topics include:

- Overview of Login Events and Processes on page 9
- Login Events on page 10
- Summary of the Login Process on page 11

Overview of Login Events and Processes

Because of the different ways that residential and enterprise subscribers connect, the login interactions between the components differ according to the type of subscriber. Because residential customers can connect by PPP, DHCP, or static IP addresses, the interactions between the SRC components differ according to the method of connection that a residential subscriber uses. However, there is only one type of login interaction—the subscriber interface login interaction—for enterprise subscribers.

Logins to plug-ins can occur during the login to the SAE or during the activation of subscriptions. For these processes, many of the interactions between the SRC components are the same regardless of the type of subscriber and the type of connection.

Related Topics

- Summary of the Login Process on page 11
- Allowing Multiple Logins from the Same IP Address on page 39

- Automatic Activation at Login on page 32
- Login Events on page 10

Login Events

Each login process begins with a login event, as described in Table 6 on page 10.

Table 6: Login Events

Login Event	Event Is Triggered When	SAE Response
AUTHINTF	An interface responds to authentication, such as authentication for a PPP session. (Supported on JUNOSe routers.)	Invokes subscriber classification script, creates subscriber session.
INTF	An interface comes up and the interface classifier script determines that the SAE should manage the interface, unless the interface comes up as a result of an authenticated PPP session. (Supported on JUNOS routing platform and JUNOSe routers.)	Invokes subscriber classification script, creates subscriber session.
ADDR	A subscriber obtains an unauthenticated IP address from the router through DHCP. (Supported on JUNOSe routers.)	Invokes subscriber classification script, creates subscriber session.
AUTHADDR	A subscriber obtains an authenticated IP address from the router through DHCP. (Supported on JUNOSe routers.)	Invokes subscriber classification script, creates subscriber session.
PORTAL	The portal API is invoked by a JSP Web page to log in a subscriber. (Supported on JUNOS routing platform and JUNOSe routers.)	Authenticate subscriber, invokes subscriber classification script, creates subscriber session.
ASSIGNEDIP	An application accesses a subscriber object for an assigned IP subscriber that is not currently loaded into memory.	Invoke subscriber classification script, creates subscriber session.

Related Topics

- Overview of Login Events and Processes on page 9
- Summary of the Login Process on page 11
- Allowing Multiple Logins from the Same IP Address on page 39
- Automatic Activation at Login on page 32

Summary of the Login Process

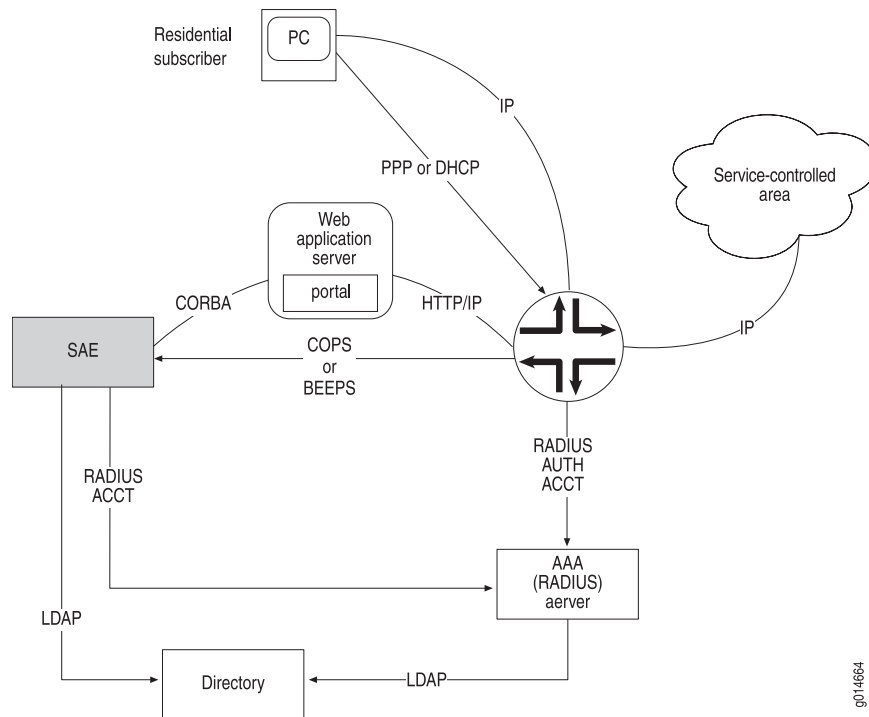
The SAE login process is summarized in the steps below. If any of the steps fail, the login process stops, and no subscriber session is created.

1. A login event occurs (see Table 6 on page 10) and triggers the login process.
2. In case of a portal login, the SAE invokes the authentication plug-ins to authenticate the request.
3. The SAE invokes the subscriber classification script and provides to the script details about the login event (for example, interface name, subscriber IP address if available, login name if available, and login event type).
4. The script sends an LDAP query that uniquely identifies a subscriber entry in the directory to the SAE.
5. The SAE loads the subscriber entry from the directory and uses the entry to create a subscriber session in memory.
6. The SAE queries all configured authorization plug-ins about whether it should allow the login.
7. The SAE completes the login process by activating the subscriber's activate-on-login subscriptions.

- Related Topics**
- Overview of Login Events and Processes on page 9
 - Residential Subscriber Login and Processes on page 11
 - Login Events on page 10
 - Enterprise Subscriber Login Process on page 26
 - Adding Residential Subscribers (SRC CLI) on page 141

Residential Subscriber Login and Processes

This section focuses on residential subscriber configurations involving authenticated PPP, DHCP, and static IP. The PPP, DHCP, and static IP cases are distinguished by the type and configuration of the networking software on the network device used to access the router. Figure 2 on page 12 shows how residential subscribers connect to SRC components.

Figure 2: Components Involved in Subscription Activation

The residential subscriber's network device (such as a computer, cellular telephone, or set-top box) connects through a layer 2 connection to the router. The network device is configured for network access with PPP or DHCP.

The router and the SAE use a RADIUS server for authentication, accounting, and optionally IP address allocation. The router can also locally manage the allocation of IP addresses to residential subscribers' PCs. A directory supporting LDAP holds the database of subscriber, service, and subscription information. Both the SAE and the RADIUS server use the directory.

Once connected to the network, the subscriber's network device exchanges IP data packets with resources in a service-controlled area. From the service provider's perspective, the resource to which access is controlled may be the network itself or content servers in the network.

The SAE manages the subscriber's IP interface on the router to control the level of access that the subscriber gets to the service-controlled area. The level of access can be anything from viewing a portal page that allows the subscriber to select a service to varying the network access speed. The subscriber can actively and instantly request access to the service-controlled area by selecting items on Web pages generated by the SAE. Selecting these items triggers the SAE to instantly reconfigure the subscriber's IP interface on the router.

The SAE communicates with JUNOSe routers through COPS messages.

The SAE communicates with JUNOS routing platforms through BEEP messages.

- Related Topics**
- Summary of the Login Process on page 11
 - PPP Subscriber Login and Service Activation on page 13
 - Automatic Activation at Login on page 32
 - Adding Residential Subscribers (SRC CLI) on page 141

PPP Subscriber Login and Service Activation

PPP subscribers access the network by using either special PPP or PPP over Ethernet software on their network access device. PPP access provides a means to configure the subscriber's network access device with several network parameters, including an IP address and a channel for transporting IP packets between the subscriber's network device and the router.

For subscribers with PPP access, logging in to the network consists of starting the PPP client, and logging out consists of stopping it. On PPP login, the router authenticates the subscriber as normal with a message to a RADIUS server. The router then notifies the SAE that there is a new IP interface on the router. The message to the SAE includes information such as the subscriber's IP address (if assigned by the router or RADIUS server), PPP login ID, and router interface ID. Using this information, the SAE retrieves the information to construct the default policies. The SAE then activates subscription policies, which are downloaded to the router and applied to the subscriber's network interface.

Subscribers can log in to the system with different accounts to different retail Internet service providers (ISPs). Subscribers use a different login ID for each account.

PPP requires special software on a network access device. The PPP software must be installed and maintained by the subscriber. The software can interfere with other applications.

Web Login for PPP Subscribers

In a PPP session, an IP address and a subscriber profile are authenticated at the same time. However, for some applications a split of subscriber profile and PPP session is useful; for example:

- Generic PPP account—An ISP could offer generic PPP login names and passwords for everybody and use Web-based login to identify subscribers.
- Device-based PPP—A PPP login may be used between a digital subscriber line (DSL) access device and a router. In this case a PPP login does not correspond to a subscriber session.
- Subaccounts with different services.

As a consequence, the Service Selection Portal (SSP) API allows creation of a Web application that:

- Allows PPP subscribers to log out—When the PPP subscriber logs out, the current subscriber session is closed, all active services are deactivated, and accounting

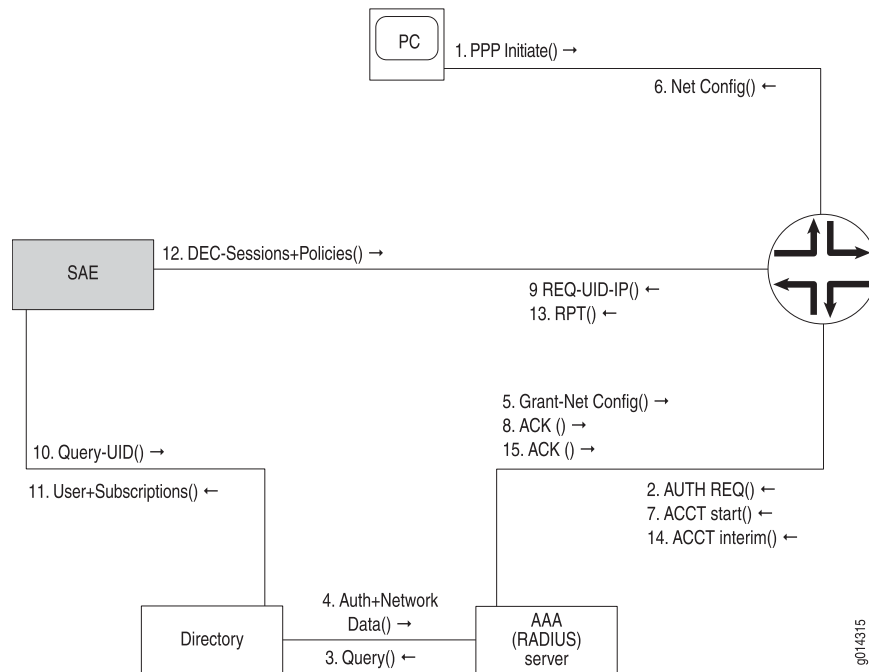
records are generated. The unauthenticated subscriber entry is then associated with the IP address of the subscriber. This process is similar to a DHCP logout.

- Forces an unauthenticated PPP subscriber (that is, a PPP subscriber account that is bound to the unauthenticated subscriber entry or to an anonymous subscriber entry) to log in—The subscriber provides a username, realm (domain), and password. Authentication is processed in the same way as a DHCP login.

PPP Login Interactions

Figure 3 on page 14 shows the interactions that take place during a PPP login.

Figure 3: PPP Login Interactions



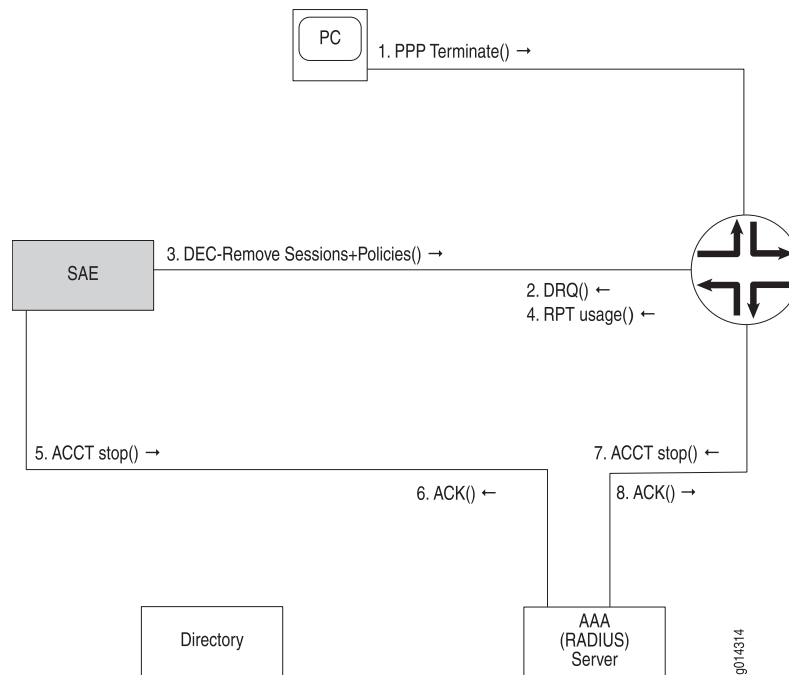
The login sequence is as follows:

1. The subscriber initiates a PPP login by starting a PPP client on his or her network device.
2. The router sends an authentication request to the RADIUS server.
3. The RADIUS server sends a user ID query to the directory.
4. The directory responds with the data (IP address for the subscriber's network device) needed to authenticate the login, and then completes the configurations of the interface on the router and on the subscriber's network device.
5. If the authentication succeeds, the RADIUS server responds to the router with a grant message, including the network configuration parameters.
6. The configurations of the PPP and IP interfaces on the router and subscriber's network device are completed.

7. The router sends an accounting start message to the RADIUS server, indicating that a subscriber session has started.
8. The RADIUS server acknowledges the accounting start message.
9. The router sends a COPS or BEEP request message to the SAE. The message includes the user ID and the IP address assigned to the IP interface on the subscriber's network device. The SAE associates the subscriber's IP address with the subscriber session so that it can associate later requests from the subscriber with this session by looking at the source IP address of the request.
10. The SAE uses the subscriber ID to look up the subscriber's data in the directory.
11. The directory responds with data about the subscriber and the associated subscriptions. This data specifies which subscriptions should be automatically activated.
12. The SAE sends a series of decision (DEC) messages to the router. These messages tell the router to attach default policies and policies for automatically activated subscriptions to the subscriber's interface. They also tell the router to store subscriber and service sessions so that if the SAE fails, the subscriber can continue using his or her active subscriptions. If the SAE fails, the router connects to a backup SAE that synchronizes all session information and then takes over management of active subscribers on the router. During the synchronization process, active sessions are not affected.
13. The router acknowledges the decision messages with a report (RPT) message.
14. If interim accounting is enabled, the router periodically sends an accounting request to the RADIUS server to store an interim accounting record.
15. The RADIUS server sends an acknowledge message to the router, acknowledging the receipt of the interim accounting record.

PPP Logout Interactions

Figure 4 on page 16 shows the interactions that take place when a subscriber logs out of a PPP session.

Figure 4: PPP Logout

The logout sequence is as follows:

1. The subscriber triggers his or her PPP software to close the PPP session with the router.
2. The router sends a COPS or BEEP delete request (DRQ) message, informing the SAE that the subscriber's IP interface is being shut down.
3. The SAE responds with decision (DEC) messages, requesting the router to remove the default and active subscription policies and sessions for the subscriber.
4. The router responds with a report (RPT) message that includes the usage data for the subscriptions that were just deactivated.
5. The SAE sends an accounting stop message to the RADIUS server, indicating that a service session has stopped. The stop message includes the usage data. (For information about service sessions, see "Subscriptions and Activations" on page 27.)
6. The RADIUS server acknowledges the accounting stop request.
7. The router sends an accounting stop message to the RADIUS server, indicating that a subscriber session has stopped.
8. The RADIUS server acknowledges the accounting stop request.

- Related Topics**
- Overview of Login Events and Processes on page 9
 - Automatic Activation at Login on page 32
 - DHCP Subscriber Login and Service Activation on page 17
 - Example: Managing Interfaces for Premium and Basic PPP and DHCP Subscribers on page 54

DHCP Subscriber Login and Service Activation

The DHCP system uses Ethernet to send data between a network device and the router. The DHCP client is built into the operating system. DHCP subscribers log in to the SAE to identify themselves, get personalized services, and select the retail ISP they want to use. Anonymous subscribers can log in to the SAE to view their account and subscription information.

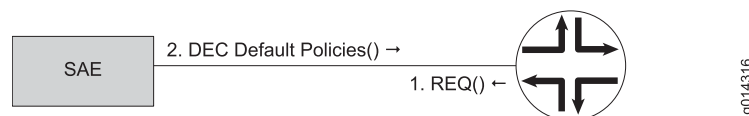
Like a subscriber with PPP access, a subscriber with DHCP access can have several accounts. The subscriber logs in to the different accounts at different times. This setup allows subscribers access to different sets of subscriptions. It supports a household in which different members share the same computer but subscribe to different services. Members of the household can get different bills for the services they use.

Subscribers can create a persistent login. In this case, the SAE stores the MAC address of the network device, along with the subscriber ID and password. This way, the network device is logged in to the subscriber account every time the device is started. Using the SAE core API, one can provide a check box on the portal page that allows the subscriber to create a persistent login.

Interface Startup

An IP interface for DHCP subscribers can come up on the router without subscribers explicitly triggering its creation by logging in. When an interface comes up, the SAE runs an interface classifier script to determine whether it should manage the interface and, if so, which default policies to apply to the interface. Thus, for DHCP subscribers, default policies are applied as soon as the IP interface on the router comes up independently of any subscriber login. Figure 5 on page 17 shows this interaction.

Figure 5: DHCP Interface Startup



The startup sequence is as follows:

1. When the IP interface on the router comes up, the router sends a COPS request (REQ) to the SAE to let it know that the new interface exists.
2. The SAE runs an interface classification script to determine whether it should manage the new interface. If the SAE manages the interface, then the SAE downloads the default policies for the interface on the router.

Initial Login

When a DHCP subscriber starts a network device for the first time, the SAE has no information about who the subscriber is and what subscriptions the subscriber has. The SAE assigns default policies and an unauthenticated subscriber profile to the

subscriber. The unauthenticated subscriber profile gives the subscriber access to services that are available without authentication.

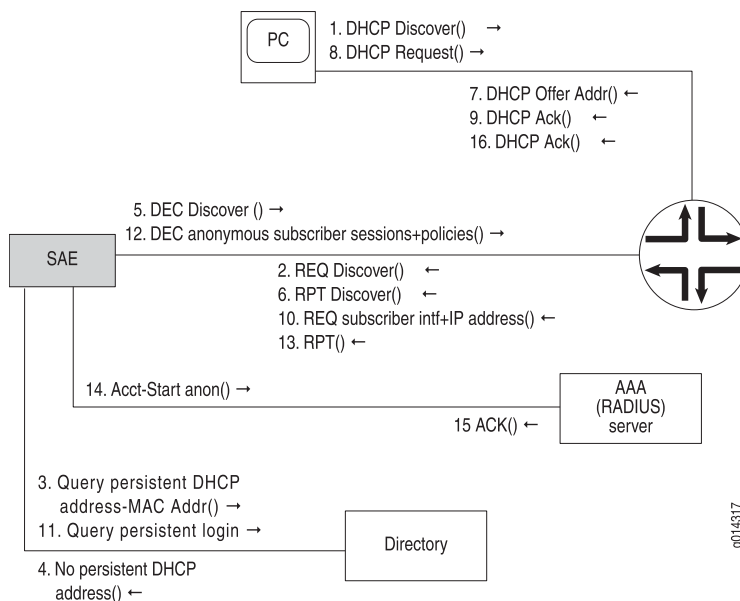
The first time a subscriber's network device starts, the router assigns an IP address to it. This address allows the subscriber access only to the SAE. The router provides this IP address for a short period of time called the lease time. After the lease time is over, the router provides a permanent IP address.

The system builds SAE applications to allow subscribers to register with the network if they are first-time subscribers of the network.

Initial DHCP Login Interactions

Figure 6 on page 18 shows the interactions that take place when a DHCP subscriber starts a network device.

Figure 6: DHCP Subscriber Initial Login



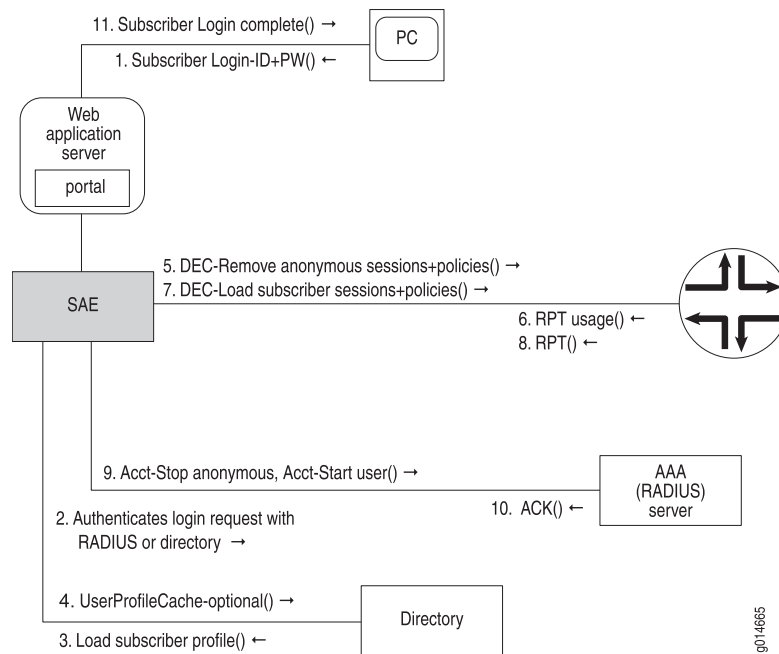
For this example, we assume that the directory responses show that there are no persistent subscriber logins. The startup sequence is as follows:

1. The DHCP client in the subscriber's network device broadcasts a discover message to the router.
2. The router acts on the discover message by sending a COPS request (REQ) message to the SAE, indicating that an IP address is about to be assigned by the local DHCP server on the local router. This request includes the MAC address of the subscriber's network device and the DHCP options sent by the client.
3. The SAE queries the directory to detect any persistent DHCP address assignments associated with the subscriber's network device. Persistent DHCP address assignments are indexed by the MAC address of the device from which they originate.

4. The directory responds with an indication that there are no persistent DHCP address assignments associated with the subscriber's network device.
5. The SAE responds to the router with a COPS decision (DEC) message, requesting the router to assign an unauthenticated address to the subscriber device.
6. The router acknowledges the address assignment decision message with a COPS report (RPT) message.
7. The router allocates and offers an IP address to the subscriber's network device.
8. The network device sends a request for the address that the router offered.
9. The router acknowledges the address request.
10. The router sends a COPS request message that includes the subscriber's interface and the assigned IP address.
11. The SAE looks up persistent logins or runs the subscriber classification script and creates a subscriber session based on the loaded subscriber profile.
12. The SAE downloads sessions for the newly logged in unauthenticated subscriber and the policies for the subscriptions that this subscriber account has configured for automatic activation. (Identification of which unauthenticated subscriber account to use is configurable in the SAE and is a function of attributes found in the original COPS request message.)
13. The router stores the sessions, applies the policies to the subscriber's IP interface, and then acknowledges the decision with a COPS report.
14. If accounting is configured for the subscriptions, the SAE sends an accounting start message to the RADIUS server.
15. The RADIUS server acknowledges the accounting message.
16. The DHCP server on the router acknowledges the DHCP renew request.

DHCP Login to Subscriber Account Interactions

Figure 7 on page 20 shows the interactions that take place when a DHCP subscriber logs in to a subscriber account. The account changes from an anonymous subscriber to an authenticated subscriber with personalized subscriptions.

Figure 7: DHCP Subscriber Login

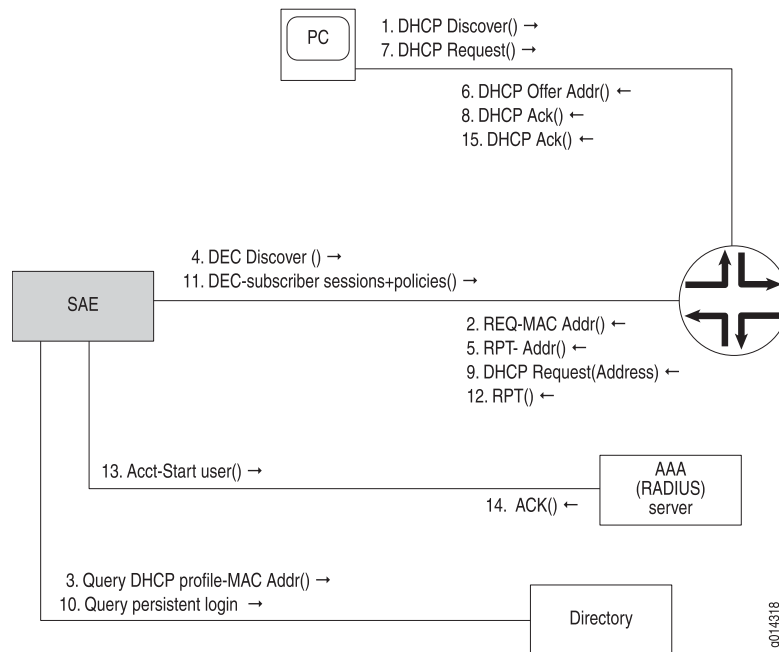
The sequence is as follows:

1. The subscriber's network device sends a request to the SAE to log in to the subscriber account with the subscriber ID and password (PW).
2. The SAE authenticates the request using the configured authentication plug-in.
3. If authentication is successful, SAE loads a subscriber profile from the directory.
4. If this is a persistent login, the SAE creates an entry in the directory in the userProfileCache object. The entry is keyed to the network device's MAC address and associates the MAC address with the subscriber ID and password. The next time the subscriber starts the device, the system automatically logs in the subscriber's account.
5. The SAE sends a COPS decision (DEC) message, instructing the router to deactivate the policies and sessions associated with the active subscriptions.
6. The router acknowledges the COPS decision message with a COPS report (RPT) message that includes usage information for the active subscriptions.
7. The SAE sends a COPS decision message to load sessions and policies for the automatically activated subscriptions for the new subscriber account.
8. The router acknowledges these decisions with COPS report messages.
9. The SAE sends the RADIUS server accounting stop messages for the subscriptions that were deactivated, and accounting start messages for the subscriptions that were activated.
10. The RADIUS server acknowledges the accounting messages.
11. The SAE responds to the subscriber's original request with a login successful message. A typical application would return a Web page that gives the subscriber the ability to activate and deactivate subscriptions.

Persistent DHCP Subscriber Login Interactions

Figure 8 on page 21 shows the interactions that take place when a DHCP subscriber starts a device on the network after having previously been logged in as a persistent subscriber.

Figure 8: Persistent DHCP Subscriber Login



The login sequence is as follows:

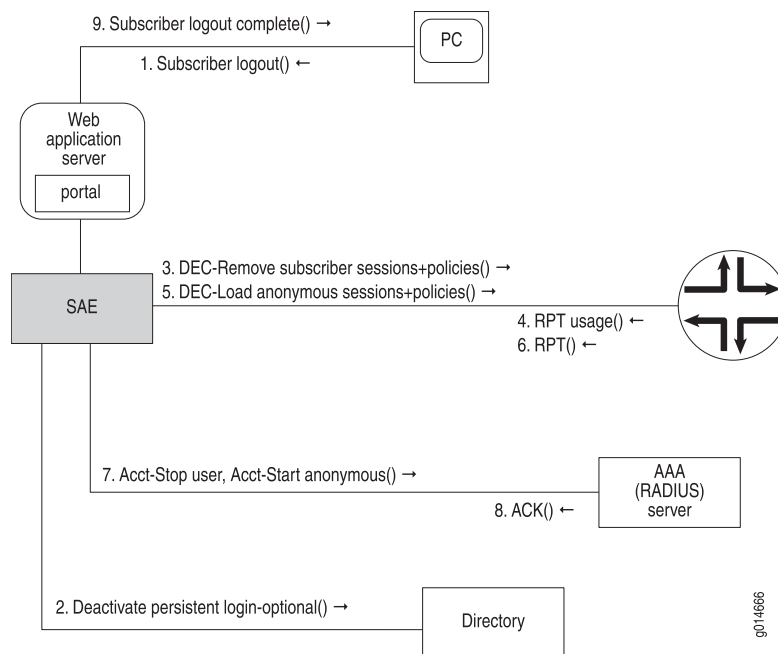
1. The DHCP client in the subscriber's network device sends a discover message to the router.
2. The router sends a COPS request (REQ) message to the SAE, informing the SAE that the router has received a DHCP discover request. The message includes the MAC address of the subscriber's network device and the DHCP options sent with the discover request.
3. The SAE queries the directory for a DHCP profile associated with the MAC address of the subscriber's network device.
4. The SAE sends the router a COPS decision (DEC) message, instructing the router to assign an IP address to the subscriber's network access device based on the information stored in the DHCP profile.
5. The router acknowledges the address assignment decision message with a COPS report (RPT) message.
6. The router allocates and offers an IP address to the subscriber's network access device.
7. The subscriber's network access device sends a request message to the router, requesting the address that was offered.

8. The router acknowledges the address request.
9. The router sends a COPS request message to the SAE that includes the subscriber's interface and the assigned IP address.
10. The SAE queries the directory for persistent logins, and the directory responds with the subscriber account information for the persistent login, including the subscriptions that are to be automatically activated.
11. The SAE starts the subscriber session and downloads session data for the subscriber account and the policies for the subscriptions that this subscriber account has configured for automatic activation.
12. The router stores the session data and applies the policies to the subscriber's IP interface. The router then acknowledges the decision message with a COPS report message.
13. If accounting is configured for the automatically activated subscriptions, then the SAE sends an accounting start message to the RADIUS server.
14. The RADIUS server acknowledges the accounting start message.
15. The router acknowledges the DHCP request messages with a DHCP acknowledge message.

DHCP Subscriber Logout Interactions

Figure 9 on page 22 shows the interactions that take place when a DHCP subscriber logs out of a subscriber account. The account changes from an authenticated subscriber to an anonymous subscriber with generic subscriptions and limited access.

Figure 9: DHCP Subscriber Logout



The logout sequence is as follows:

1. The subscriber's network device sends a request to the SAE to log out of its current subscriber session.
2. The subscriber may request to deactivate persistent login. If the subscriber deactivates persistent login, the SAE deletes the entry in the directory. If the subscriber does not deactivate the persistent login, then the account is automatically logged in the next time the same network device is started.
3. The SAE sends a COPS decision (DEC) message to the router, instructing the router to remove the sessions and policies associated with the active subscriptions.
4. The router responds with a COPS report (RPT) message that includes the usage information for the deactivated subscriptions.
5. The SAE sends a COPS decision message to add sessions and policies for the automatically activated subscriptions for the anonymous account to which the subscriber has switched.
6. The router acknowledges the COPS decision message by sending a COPS report message to the SAE.
7. The SAE sends the RADIUS server accounting stop messages for the subscriptions that were deactivated, and accounting start messages for the subscriptions that were activated.
8. The RADIUS server acknowledges these accounting messages.
9. The SAE responds to the subscriber's logout request, showing that the logout is complete.

- Related Topics**
- Overview of Login Events and Processes on page 9
 - Subscriptions and Activations on page 27
 - Classifying DHCP Subscribers (SRC CLI) on page 68
 - PPP Subscriber Login and Service Activation on page 13
 - Example: Managing Interfaces for Premium and Basic PPP and DHCP Subscribers on page 54

Static IP Subscribers

The SAE supports residential subscribers who use statically assigned IP addresses. Statically assigned means that the network does not create events that contain information about the IP address of the subscriber. The SAE can handle the case in which a router interface is dedicated to one subscriber. This subscriber can be a single PC or multiple PCs that are managed by the same household.

Single PC, IP Address Known

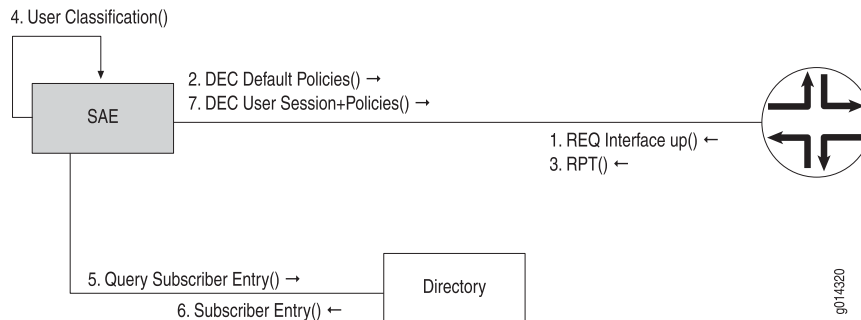
See Figure 10 on page 24.

1. When the interface dedicated to the subscriber comes up, the router sends a COPS or BEEP request (REQ) message to the SAE. The SAE calls the interface classification script to determine whether the interface is being managed and which default policies are applied.
2. The SAE sends a decision (DEC) message to the router, requesting that the router attach the selected default policies.
3. The router acknowledges the decision message with a report message.
4. The SAE calls the subscriber classification script to determine whether a subscriber session needs to be started. The subscriber classification script responds with an LDAP query.
5. The SAE uses the LDAP query to look up a subscriber entry in the directory.
6. The directory responds with data about the subscriber and the associated subscriptions. The IP address assigned to the subscriber can be part of the data returned from the directory. If the IP address cannot be stored in the directory, it is also possible to integrate the SAE with an external data source (for example, a database maintained by an existing provisioning system), to look up the IP address of the subscriber.

As in the PPP case, the SAE associates the subscriber session with the IP address so it can handle later requests by looking up the source IP address of the HTTP request.

7. The SAE sends decision messages that install policies for automatically activated subscriptions.

Figure 10: Static IP Subscriber Login



Subscriber IP Address Not Known

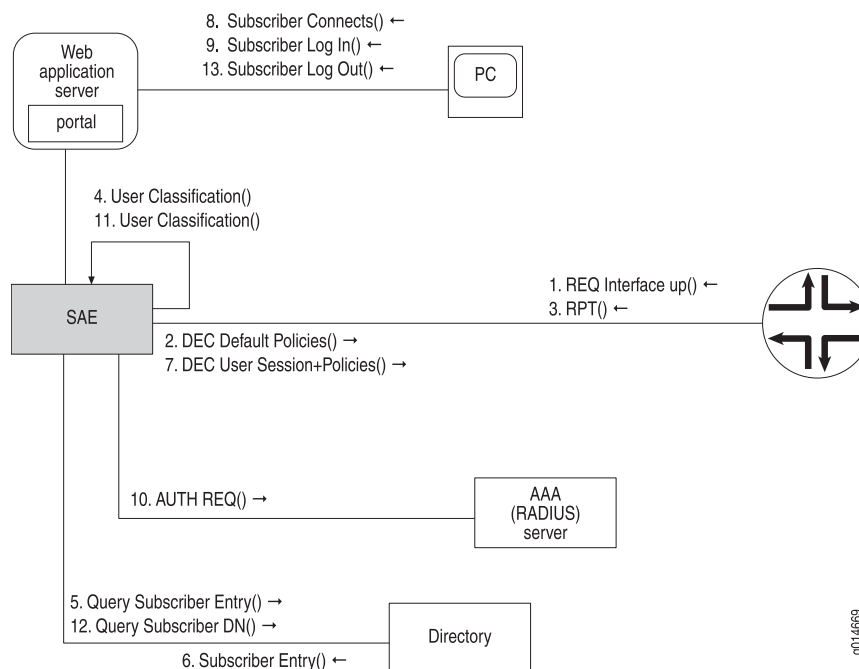
See Figure 11 on page 26.

1. When the interface dedicated to the subscriber comes up, the router sends a BEEP or COPS request (REQ) message to the SAE. The SAE calls the interface classification script to determine whether the interface is being managed and which default policies are applied.
2. The SAE sends a decision (DEC) message to the router, requesting that the router attach the selected default policies.

3. The router acknowledges the decision message with a report (RPT) message.
4. The SAE invokes the subscriber classification script to determine whether a subscriber session needs to be started. The subscriber classification script responds with an LDAP query.
5. The SAE uses the LDAP query to look up a subscriber entry in the directory.
6. The directory responds with data about the subscriber and the associated subscriptions.

The SAE associates the subscriber session with the DN of the subscriber entry so that later requests can be handled. One consequence of associating the subscriber entry with the DN is that it is not possible to have more than one subscriber session for a single DN active at the same time.

7. The SAE sends decision messages that install policies for automatically activated subscriptions.
8. The subscriber connects to the portal. Because the IP address of the subscriber is not associated with a subscriber session, a login page is displayed instead.
9. The subscriber provides a username and password.
10. The SAE authenticates the request (for example, by using the RADIUS authentication plug-in) and calls the subscriber classification script.
11. The subscriber classification script returns an LDAP query. The SAE uses the query to look up the DN of the subscriber entry in the directory.
12. The SAE uses the DN returned from the directory to find a subscriber session and associates it with the IP address of the HTTP request. The SAE handles subsequent accesses to the portal by looking up the IP address of the HTTP request.
13. The subscriber logs out from the SAE. The SAE does not change the subscriber session associated with the DN of the subscriber, but removes the association of the subscriber IP address with the subscriber session.

Figure 11: Subscriber IP Address Not Known

- Related Topics**
- Overview of Subscribers on page 3
 - Overview of Classification Scripts on page 43
 - Classifying Interfaces (SRC CLI) on page 49
 - Allowing Multiple Logins from the Same IP Address on page 39

Enterprise Subscriber Login Process

Enterprise subscribers may connect through any access method. Any of the events described in Table 6 on page 10 can initiate an enterprise login.

Interface Startup

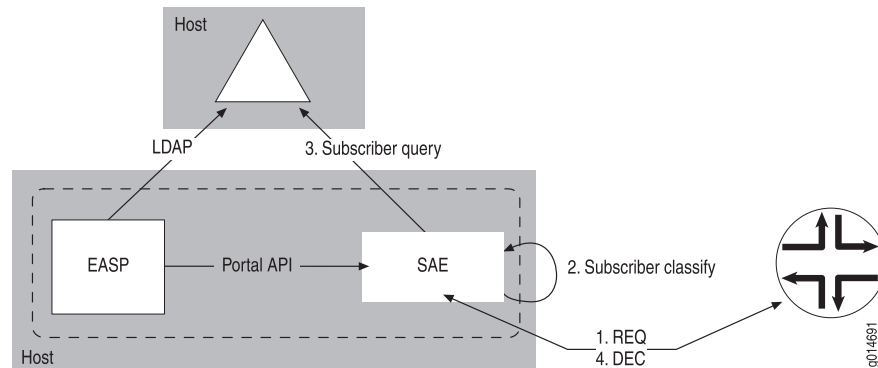
When a router interface comes up, the router sends a message to the SAE with information about that interface.

The SAE classifies the subscriber to determine the default interface policies. An SAE subscriber classification rule matches the attributes of the interface and describes how to formulate an LDAP query that retrieves the access entry in the directory that corresponds to the router interface.

Based on the response from the directory, the SAE creates a subscriber session and associates it with the DN of the access entry in the directory. The SAE then sends the router a message to install all the policies for subscriptions for the access line that are set to administratively active.

Figure 12 on page 27 shows the stages involved in activating an enterprise subscriber session.

Figure 12: Enterprise Subscriber Session Activation



- Related Topics**
- Summary of the Login Process on page 11
 - Residential Subscriber Login and Processes on page 11
 - Automatic Activation at Login on page 32
 - Adding Enterprises (SRC CLI) on page 145
 - Configuring Administrative Information for Enterprise Subscribers (SRC CLI) on page 147


Subscriptions and Activations

Each subscriber purchases a set of services; this purchase is known as a subscription. Information about the subscriptions is stored in the directory and is used by a residential service selection portal application to generate controls that enable the subscriber to:

- Activate and deactivate subscriptions.
- Subscribe to services.
- Configure subscriptions to be automatically activated.

The service selection application can be either a Web application or an API. When the service selection application is a Web application, the controls are Web pages with buttons and links to click on (see Figure 13 on page 28 and Figure 14 on page 29). However, the service selection application provides an open API that makes it possible to build applications that are controlled by mechanisms other than Web pages. For instance, customers can build service selection applications that are controlled by applications running in the system tray area of the Windows task bar. This deployment consolidates the control of subscribers' active network services and the speed of their Internet connection, along with their control of other aspects of their PC, such as the clock settings and audio volumes.

Figure 13: Service Activation Page



Hello Jane User

HomeLogoutContact us

Service Selection Portal ▶

▶ Services

▶ Usage

▶ Account

▶ Schedules

▶ Subscribe

▶ Register


▶ Unregister

Search

Services


You can start or stop a service by clicking on the circle in the "Status" column. A green circle (✔) means the service is currently on. A red circle (●) means the service is currently off.

You can persistently activate a service by clicking on the check box in the "Persistent" column. Persistently activated services are automatically activated when you login to the portal.



Internet

Service Description	Status	Password required	Persistent	Price
Example for rate limited internet (requires matching default policies)	✔		<input type="checkbox"/>	N/A



Copyright © 1999-2003 Juniper Networks

Figure 14: Subscription Activation Page

virneo
The network that keeps you surfing

Hello Jane User

Home Logout Contact us

Service Selection Portal

- Services
- Usage
- Account
- Schedules
- Subscribe**
- Register
- Unregister

Search

Subscribe

All available services are listed below.

It may take a minute for your new subscriptions to take effect.

Internet	Overwrite	Security	Video	Quality of Service	Audio	News	Denial of Service	
Service Name	Service description						Subscribed	Unsubscribed
Internet-Bronze	Example for rate limited internet (requires matching default policies)						<input checked="" type="radio"/>	<input type="radio"/>
Internet-Gold	Example for rate limited internet (requires matching default policies)						<input type="radio"/>	<input checked="" type="radio"/>
Internet-Silver	Example for rate limited internet (requires matching default policies)						<input type="radio"/>	<input checked="" type="radio"/>

OK Cancel

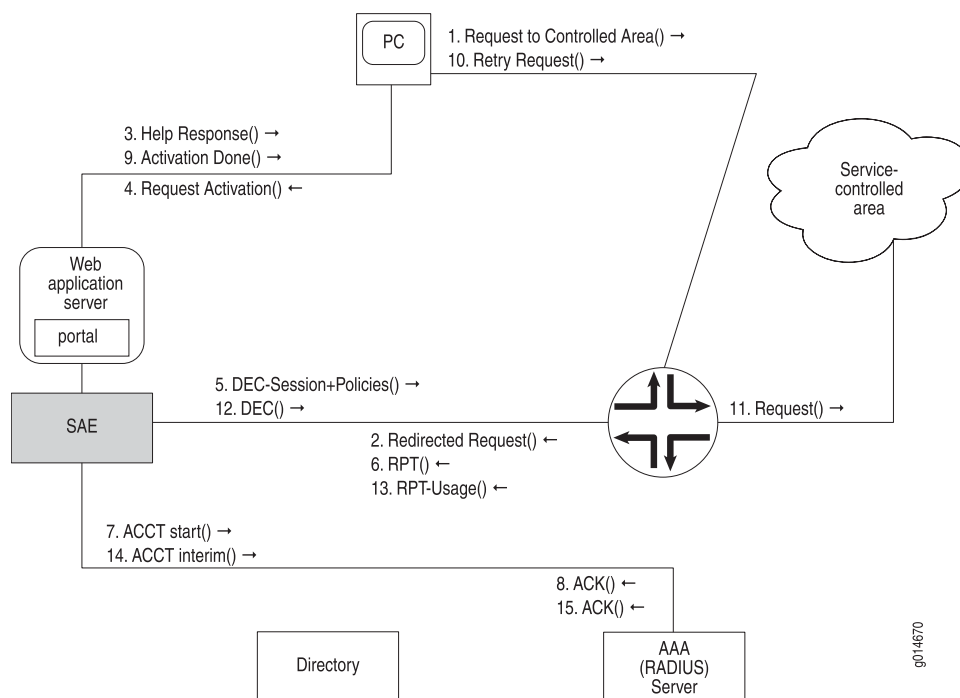
virneo

Copyright © 1999-2003 Juniper Networks

Many of the activation and deactivation interactions work in the same way, whether the subscriber is a residential subscriber or an enterprise subscriber. However, some interactions apply only to enterprise subscribers.

Subscription Activation Interactions

Clicking a button on the Web page to activate a service session causes the SAE to download the policies associated with the service to the subscriber's IP interface on the router. Figure 15 on page 30 shows the interactions among the components shown in Figure 2 on page 12 during the activation process. This scenario assumes that the subscriber has already logged in.

Figure 15: Subscription Activation

The activation sequence is as follows:

1. Before the subscription is activated, the subscriber makes a request to the corresponding subscription resource in the service-controlled area.
2. A default policy that matches the request on the router causes the router to redirect the request to the SAE.
3. The SAE responds to the request with a help desk Web page, requesting that the subscriber activate the subscription before trying to access the resource.
4. The subscriber clicks a button on the service selection portal Web page, requesting the activation of the subscription.
5. The SAE sends a COPS or BEEP decision (DEC) message to the router, requesting the installation of policies for the subscription on the subscriber's IP interface on the router, as well as service session information.

At start time, the SAE loads all services and policy templates from the directory. At activation time, the policy templates for the service are instantiated with values that are determined at activation, such as the subscriber's IP address. The router stores session information so that if the SAE fails, the subscriber can continue using his or her active subscriptions. If the SAE fails, the router connects to a backup SAE. The backup SAE synchronizes all session information and then takes over management of all active subscribers on the router.

6. The router responds with a report (RPT) message acknowledging the decision message.
7. The SAE sends an accounting start message to the RADIUS server.
8. The RADIUS server acknowledges the accounting start message.

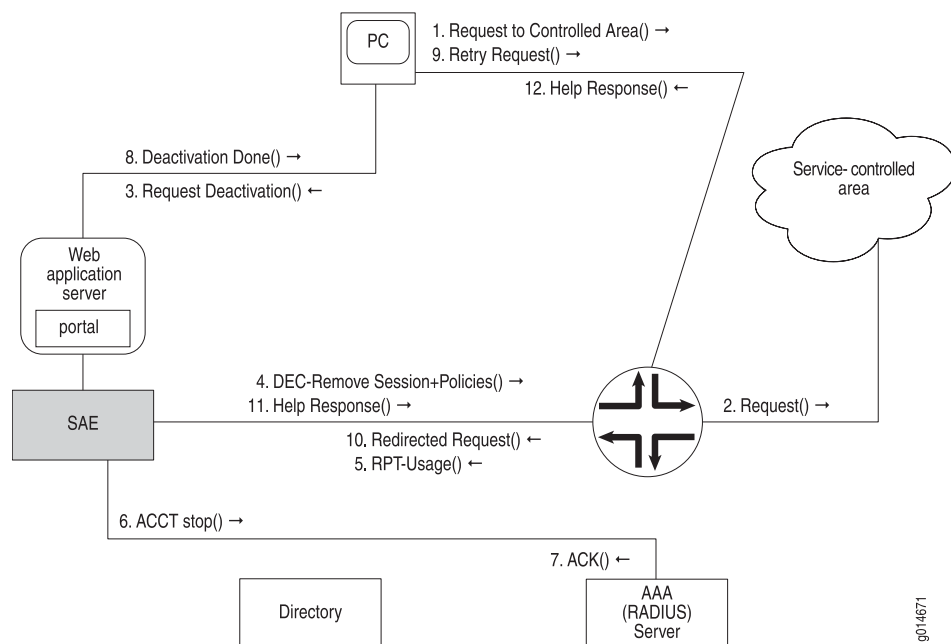
9. The SAE responds to the subscriber's activation request, indicating that the subscription is active.
10. The subscriber may now retry the request for access to the controlled resource.
11. This time, the request to the controlled resource matches the policy from the newly activated subscription, so the router allows the request to be routed normally. Depending on the policy, the router may also apply QoS processing.
12. If interim accounting is enabled, the SAE periodically sends a decision message requesting usage data.
13. The router responds with a report message that contains usage data for the subscription. The usage data consists of the number of bytes and packets that the policies processed for the subscription.
14. The SAE stores the usage data in interim accounting records in the RADIUS server.
15. The RADIUS server acknowledges the interim accounting record.

Subscription Deactivation Interactions

Clicking a button on the Web page to deactivate a service causes the SAE to request that the router remove the policies for the service from the subscriber's IP interface on the router.

Figure 16 on page 31 shows the interactions among the components shown in Figure 2 on page 12 during the subscription deactivation process. This scenario assumes that the subscriber has already logged in.

Figure 16: Subscription Deactivation



The deactivation sequence is as follows:

1. The subscriber sends a request to deactivate a subscription to a resource in the service-controlled area.
2. The request matches a policy that allows the request to be forwarded to the resource in the service-controlled area.
3. The subscriber clicks on a field on a Web page to request that the SAE deactivate the subscription.
4. As a result, the SAE sends a COPS or BEEP decision (DEC) message to the router to remove policies for the subscription from the subscriber interface and the service session from memory.
5. The router acknowledges the decision message with a report (RPT) message that contains service usage. The usage is the number of bytes and packets that the policies processed for the subscription.
6. An accounting stop record that includes the subscription usage information is written in the RADIUS server.
7. The RADIUS server acknowledges the accounting message.
8. The SAE sends a message to the subscriber, informing the subscriber that the subscription has been deactivated.
9. Because the policy for the subscription was removed from the subscriber interface on the router, any request for access is directed to the SAE.
10. The subscriber may now retry to request access to the controlled resource.
11. As was the case before the subscription was activated, the SAE generates a help desk Web page response that is relayed to the subscriber.

- Related Topics**
- Overview of Subscriptions on page 4
 - Automatic Activation at Login on page 32
 - Residential Subscriber Login and Processes on page 11
 - Configuring Subscriptions (SRC CLI) on page 153
 - Configuring Accesses (SRC CLI) on page 155

Automatic Activation at Login

An activate-on-login subscription is a subscription that is configured to start every time the subscriber logs in.

A manual subscription is a subscription that is configured to start only by an action from the subscriber.

For example, residential subscriber Elizabeth has designated her high-speed subscription to automatically activate every time she logs in. On the other hand, her video subscription is not activated unless she activates it by clicking a button on a portal page. It is possible to integrate the SAE with a video-on-demand server so that the video service is automatically activated when Elizabeth logs in. This type of configuration ensures access to the server and to QoS for the video stream. When

the video stream is finished, the video-on-demand server triggers the SAE to stop the video service.

Residential subscriber Robert is interested in streaming audio. He sets his subscriptions so that regular-speed service, along with his subscription to an audio service, is automatically activated every time he logs in.

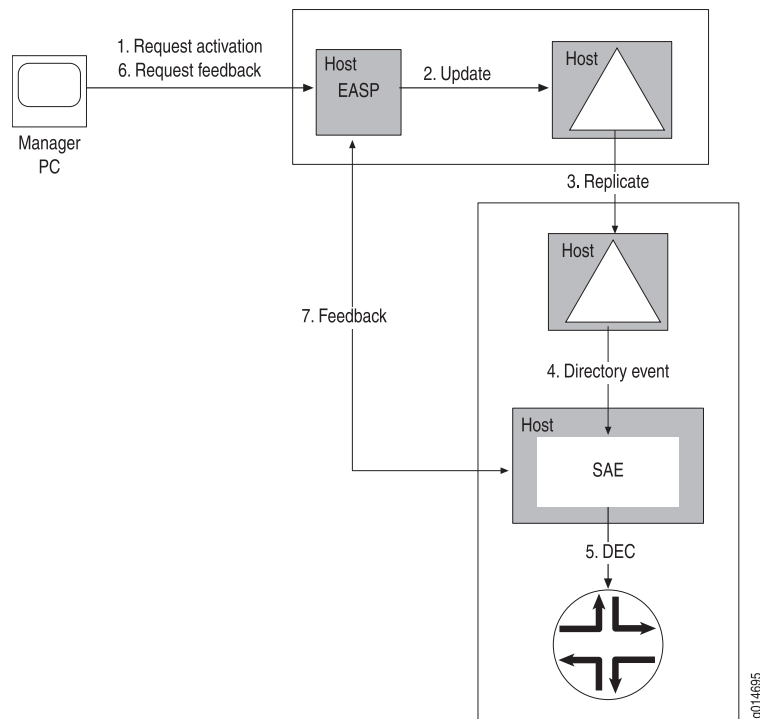
Enterprise-Specific Remote Session Activation

When a subscription is set for automatic activation through the Web interface, a service session request message is sent from the manager's PC to the Enterprise Manager Portal. The Enterprise Manager Portal writes this request to the directory, and the directory eventing system (DES) notifies the SAE affected by this request of the directory event. The SAE then sends a COPS or BEEP decision message to the router to download the policies for the activated subscription.

The enterprise manager must explicitly request feedback to see whether the session succeeded and what the operational values for the service parameters actually are. To do this, the enterprise manager sends a feedback request to the Enterprise Manager Portal. To process this request, the Enterprise Manager Portal sends a feedback request to the remote SAE managing the access through CORBA and returns the response to the enterprise manager's browser.

Figure 17 on page 33 shows the sequence of messaging events that occur between the manager PC, the Enterprise Manager Portal, the master and shadow directories, the remote SAE, and the router.

Figure 17: Remote Session Activation Sequence



- Related Topics**
- Overview of Login Events and Processes on page 9
 - Summary of the Login Process on page 11
 - Subscriptions and Activations on page 27
 - Residential Subscriber Login and Processes on page 11
 - Overview of Managers on page 6

Chapter 3

Configuring Subscriber-Related Properties on the SAE (SRC CLI)

- Configuring the Length of Time MAC Addresses Remain in SAE Cache on page 35
- Identifying a Profile for Unauthenticated Subscribers on page 36
- Configuring Interim Accounting for Services and Subscribers on page 37
- Avoiding Overcharges for Sessions That Time Out on page 38
- Allowing Multiple Logins from the Same IP Address on page 39
- Authenticating Registered Username/Password Pairs on page 40
- Configuring Timers for Session Reactivation on page 40

Configuring the Length of Time MAC Addresses Remain in SAE Cache

When a DHCP subscriber transitions from an authenticated IP address to an unauthenticated IP address or vice-versa, the SAE:

1. Logs out the subscriber associated with the original IP address.
2. Caches the subscriber profile in the in-memory cache, indexed by the DHCP subscriber's MAC address.
3. Waits until the DHCP subscriber with the cached MAC address obtains its new IP address, and then logs in the subscriber and associates it with the new IP address.

The period during which the subscriber profile remains in the in-memory cache can last until the DHCP lease time for the original address. If something happens during this period—for example, the subscriber turns off the client computer—the subscriber profile remains in the SAE's in-memory cache forever. When a new IP address is assigned to the same DHCP client, problems can occur. To avoid such problems, entries in the in-memory cache are removed after a configurable amount of time.

Configure the amount of time that entries remain in cache to be greater than the time required for a DHCP subscriber to transition from an unauthenticated IP address to an authenticated IP address or vice versa. The time required for a DHCP subscriber to transition from one IP address to another depends on the lease times configured on the JUNOS router and the instructions given to the subscriber on the Web portal, such as reboot your PC now.

Use the following configuration statement to configure the length of time that a subscriber profile remains in the SAE's in-memory cache:

```
shared sae configuration driver {
    mac-cache-expiration mac-cache-expiration;
}
```

To configure the amount of time that subscriber profiles remain in the SAE's in-memory cache:

1. From configuration mode, access the SAE driver configuration statement.

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

2. Specify the amount of time that subscriber profiles remain in the SAE's cache.

```
[edit shared sae configuration driver]
user@host# set mac-cache-expiration mac-cache-expiration
```

3. (Optional) Verify your configuration.

```
[edit shared sae configuration driver]
user@host# show mac-cache-expiration
mac-cache-expiration 1800;
```

- Related Topics**
- Configuring the Length of Time That MAC Addresses Remain in SAE Cache (C-Web Interface)
 - DHCP Subscriber Login and Service Activation on page 17

Identifying a Profile for Unauthenticated Subscribers

The SAE uses an unauthenticated subscriber profile as a transitional profile for subscribers who are not logged in to the SAE. For example, if a subscriber logs out of the SAE using the API method `Subscriber.logout()`, an unauthenticated subscriber session is created. The unauthenticated subscriber profile must exist and can be subscribed to services available for unauthenticated subscribers. The portal implementation determines whether unauthenticated (anonymous) subscribers can access the portal.

Use the following configuration statement to specify an unauthenticated subscriber profile.

```
shared sae configuration driver {
    unauthenticated-subscriber-dn unauthenticated-subscriber-dn
}
```

To specify an unauthenticated subscriber profile:

1. From configuration mode, access the SAE driver configuration statement.

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

2. Specify a subscriber profile for unauthenticated access to the portal.

```
[edit shared sae configuration driver]
user@host# set unauthenticated-subscriber-dn unauthenticated-subscriber-dn
```

3. (Optional) Verify your configuration.

```
[edit shared sae configuration driver]
user@host# show unauthenticated-subscriber-dn
unauthenticated-subscriber-dn
uniqueID=unauthentication,ou=local,RetailerName=default,o=Users,<base>;
```

- Related Topics**
- Identifying a Profile for Unauthenticated Subscribers (C-Web Interface)
 - Authenticating Registered Username/Password Pairs on page 40
 - Overview of Subscribers on page 3

Configuring Interim Accounting for Services and Subscribers

You can enable and disable interim accounting and set intervals between interim accounting messages for services and subscribers. These settings apply to all subscriber sessions and service sessions unless you override these settings for specific services by configuring an accounting interim interval in the value-added service configuration.

Use the following configuration statements to configure interim accounting.

```
shared sae configuration interim-accounting {
  service-interim-accounting;
  service-interim-interval service-interim-interval ;
  subscriber-interim-accounting;
  subscriber-interim-interval subscriber-interim-interval ;
}
```

To set up interim accounting:

1. From configuration mode, access the configuration statement for interim accounting.

```
user@host# edit shared sae configuration interim-accounting
```

2. (Optional) Enable service interim accounting.

```
[edit shared sae configuration interim-accounting]
user@host# set service-interim-accounting
```

3. Specify the interval between service interim accounting messages.

```
[edit shared sae configuration interim-accounting]
user@host# set service-interim-interval service-interim-interval
```

4. (Optional) Enable interim accounting for subscribers.

```
[edit shared sae configuration interim-accounting]
user@host# set subscriber-interim-accounting
```

5. Specify the interval between subscriber interim accounting messages.

```
[edit shared sae configuration interim-accounting]
user@host# set subscriber-interim-interval subscriber-interim-interval
```

6. Verify your configuration.

```
[edit shared sae configuration interim-accounting]
user@host# show
service-interim-accounting;
service-interim-interval 900;
subscriber-interim-accounting;
subscriber-interim-interval 900;
```

- Related Topics**
- Configuring Interim Accounting for Services and Subscribers (C-Web Interface)
 - SAE Accounting
 - Overview of Configuring Subscribers and Subscriptions on page 135

Avoiding Overcharges for Sessions That Time Out

When an idle timeout terminates a session, you can set up the SAE to reduce the session time reported in the accounting stop message by the idle time. This way the session time is accurately reported to avoid overcharges for the session.

Use the following configuration statement to configure the length of time that a subscriber profile remains in the SAE's in-memory cache:

```
shared sae configuration idle-timeout {
  adjust-session-time;
}
```

To adjust the session time:

1. From configuration mode, access the SAE idle timeout configuration statement.

```
user@host# edit shared sae configuration idle-timeout
```

2. Enable when an idle timeout terminates a session, the session time reported in the accounting stop message is reduced by the idle time.

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

3. (Optional) Verify your configuration.

```
[edit shared sae configuration idle-timeout]
user@host# show
adjust-session-time;
```

- Related Topics**
- Avoiding Overcharges for Sessions That Time Out (C-Web Interface)
 - Configuring Timers for Session Reactivation on page 40
 - Tracking and Controlling Subscriber and Service Sessions with SAE APIs

Allowing Multiple Logins from the Same IP Address

You can specify whether the SAE allows a login from the same IP address without requiring that the previous session logs out first.

- If you enable this setting, the SAE logs in the new subscriber session and automatically logs out the previous session.
- If you disable this setting, the SAE denies login requests if a subscriber session for an IP address is active.

Use the following configuration statement to specify whether or not the SAE allows multiple logins from the same IP address:

```
shared sae configuration subscriber-sessions {
  allow-same-ip-login;
}
```

To specify whether the SAE allows a login from the same IP address without requiring that the previous session logs out first:

1. From configuration mode, access the subscriber sessions statement.

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

2. Enable or disable whether the SAE allows a login from the same IP address without requiring that the previous session logs out first.

```
[edit shared sae configuration subscriber-sessions]
user@host# set allow-same-ip-login
```

3. (Optional) Verify your configuration.

```
[edit shared sae configuration subscriber-sessions]
user@host# show
adjust-session-time;
```

- Related Topics**
- Allowing Multiple Logins from the Same IP Address (C-Web Interface)
 - Static IP Subscribers on page 23

Authenticating Registered Username/Password Pairs

You can specify whether the application programming interface (API) method `registerLoginCredentials` authenticates the registered username/password or creates the registration without authentication. You should enable this setting if your authentication server does not allow authentication while a session for the authenticated username is active.

Use the following configuration statement to specify whether or not registered username/password pairs are authenticated:

```
shared sae configuration login-registration {
    registration-authentication;
}
```

To specify whether or not registered username/password pairs are authenticated:

1. From configuration mode, access the subscriber sessions statement.

```
user@host# edit shared sae configuration login-registration
```

2. Enable or disable whether registered username/password pairs are authenticated.

```
[edit shared sae configuration login-registration]
user@host# set registration-authentication
```

3. (Optional) Verify your configuration.

```
[edit shared sae configuration login-registration]
user@host# show
registration-authentication;
```

- Related Topics**
- Authenticating Registered Username/Password Pairs (C-Web Interface)
 - Identifying a Profile for Unauthenticated Subscribers on page 36
 - Tracking and Controlling Subscriber and Service Sessions with SAE APIs

Configuring Timers for Session Reactivation

If a service session fails unexpectedly, the SAE tries to start the session again in the background. You can change how many times the SAE tries to activate the session and the interval between these attempts. In most instances, you do not need to change the default values.

Use the following configuration statements to configure background session reactivation behavior

```
shared sae configuration service-activation {
    retry-time retry-time ;
    retry-limit retry-limit ;
}
```


To configure session reactivation behavior:

1. From configuration mode, access the service activation statements.

```
user@host# edit shared sae configuration service-activation
```

2. Configure the number of times the SAE tries to activate a service session if activation fails or to deactivate a service session if deactivation fails.

```
[edit shared sae configuration service-activation]
user@host# set retry-limit retry-limit
```

3. Configure the time between attempts to activate a service session if activation fails or to deactivate a service session if deactivation fails.

```
[edit shared sae configuration service-activation]
user@host# set retry-time retry-time
```

4. (Optional) Verify your configuration.

```
[edit shared sae configuration service-activation]
user@host# show
retry-time 60;
retry-limit -1;
```

- Related Topics**
- Configuring Timers for Session Reactivation (C-Web Interface)
 - Avoiding Overcharges for Sessions That Time Out on page 38
 - Subscriptions and Activations on page 27
 - Automatic Activation at Login on page 32

Chapter 4

Classifying Interfaces and Subscribers (SRC CLI)

- Overview of Classification Scripts on page 43
- Overview of Configuring Classification Scripts on page 46
- Classifying Interfaces (SRC CLI) on page 49
- Example: Managing Interfaces for Premium and Basic PPP and DHCP Subscribers on page 54
- Example: Managing Specific Interfaces on page 55
- Example: Managing Interfaces by Using the Interface Description on page 55
- Classifying Subscribers (SRC CLI) on page 56
- Sending DHCP Options to the JUNOS Router on page 63
- Subscriber Classification Targets on page 64
- Example: Subscriber Classification Scripts for Static IP Subscriber on page 65
- Example: Subscriber Classification Scripts Using a Subscriber Group on page 66
- Example: Subscriber Classification Scripts for Enterprise Subscribers on page 66
- Example: Creating Router Interface Subscriber Session on page 67
- Example: Activating Services for a Group of Subscriber Sessions on page 68
- Classifying DHCP Subscribers (SRC CLI) on page 68
- Syntax for DHCP Classification Targets on page 72
- Selecting DHCP Parameters on page 73
- DHCP Options Supported on the SAE on page 74
- Creating DHCP Profiles (SRC CLI) on page 77

Overview of Classification Scripts

The SAE uses classification scripts to determine whether it manages router interfaces, to select default policies, to find subscriber profiles, and to choose DHCP profiles. The SAE has three classification scripts:

- Interface classification script—When a subscriber's IP interface comes up on the router, the router sends the subscriber's login and interface information to the SAE. The SAE runs the interface classification script to determine whether the SAE manages the interface and if so, what default policies to send to the router.

- Subscriber classification script—If the SAE is managing the interface, the SAE uses the login and interface information that the router sends to run the subscriber classification script to determine which subscriber profile to load into memory.
- DHCP classification script—For DHCP subscribers, the SAE runs DHCP classification scripts to choose DHCP profiles.

How Classification Scripts Work

Classification scripts consist of *targets* and *conditions*.

- A target is the result of the classification script. For example, the result of subscriber classification scripts is an LDAP search string that is used to find a unique subscriber profile. The result of interface classification scripts is a policy group.
- Conditions are match criteria. The script attempts to match conditions in the script with information sent from the router. For example, match conditions for a subscriber classification script might be login type or domain name. Match conditions for an interface classification script could be interface IP address or interface description.

Each script can have multiple targets, and each target can have multiple conditions. When an object needs classification, the script processes the targets in turn. Within each target, the script processes conditions sequentially. When it finds that the classification conditions for a target match, it returns the target to the SAE. If the script does not find any targets that can be matched, the classifier engine returns a no-match message to the SAE.

Because classification scripts examine conditions sequentially as the conditions appear in the script, you should put more specific conditions at the beginning of the script and less specific conditions at the end of the script.

Interface Classification Scripts

When a subscriber's IP interface comes up on the router, the router sends the subscriber's login and interface information to the SAE. For example, the router might send the following information:

```
IP address=0.0.0.0
Virtual router name=default@erx5_ssp58
Interface name=FastEthernet3/1.1
PPP login name (PPP)=pebbles@virneo.net
User IP address (PPP)=192.168.55.5
Interface speed=100000000
Interface description=P3/1.1
Interface alias=1st pppoe int
RADIUS class=null
```

The SAE invokes the interface classification script and provides to the script the information that it received from the router. The script engine matches the information sent from the router to the conditions in the interface classification script. The script examines each condition in sequential order to find a match.

- If it finds a match, the script processing stops, and the target for that condition is returned to the SAE. The target is the path of a policy group. This policy group is the default policy. The SAE installs the policy on the interface and begins managing the interface.
- If it does not find a match, the script sends a no-match message to the SAE. The SAE does not manage the interface; that is, the policies installed through RADIUS or the CLI remain in effect. The SAE does not install policies.

Subscriber Classification Scripts

When the SAE begins managing an interface, it determines whether a subscriber is associated with the interface by running the subscriber classification script. The SAE also runs the subscriber classification script when certain login events occur. See “Login Events” on page 10 for a description of login event types.

To find the matching subscriber profile, the SAE uses interface information that it received from the router when the interface became operational (for example, virtual router name, interface name, interface alias). It also uses login information that it received from the router or the portal application when the subscriber attempted to log in (for example, subscriber IP address, login name, or login event type).

When the SAE runs the subscriber classification script, the script engine matches the information sent from the router to the conditions in the subscriber classification script. The script examines each condition in sequential order to find a match.

- If it finds a match, the script processing stops, and the target for the matching condition is returned to the SAE. The target is an LDAP query that uniquely identifies a subscriber profile. The SAE loads the subscriber entry and uses the entry to create a subscriber session in memory.
- If it does not find a match, the script sends a no-match message to the SAE. The SAE does not load a subscriber session onto the interface, and services cannot be activated for this session.

DHCP Classification Scripts

DHCP classification scripts choose DHCP profiles. See “Assigning DHCP Addresses to Subscribers” on page 84 for information about how DHCP classification scripts are used.

Sharing Information Among Classification Scripts

In many instances, the same classification rule may appear in different classification scripts. You can reuse the same information in different scripts by configuring the information in one script and including that information in another script. Interface, subscriber, and DHCP classification scripts all let you include another script.

- Related Topics**
- Overview of Configuring Classification Scripts on page 46
 - Classifying Interfaces (SRC CLI) on page 49
 - Classifying Interfaces (C-Web Interface)

- Classifying Subscribers (SRC CLI) on page 56
- Interface Classification Conditions on page 52
- Classifying DHCP Subscribers (SRC CLI) on page 68

Overview of Configuring Classification Scripts

Classification scripts are organized into rules. Each rule has a target and one or more match conditions. For example:

Subscriber Classifiers

```
subscriber-classifier {
.
.
.
rule rule-2 {
    target <-unauthenticatedUserDn->;
    condition {
        "loginType == \"ADDR\"";
        "loginType == \"AUTHADDR\"";
    }
}
}
```

DHCP Classifiers

```
dhcp-classifier {
.
.
.
rule rule-2 {
    target cn=default,<-dhcpProfileDN->;
    condition {
        1;
    }
}
}
```

Interface Classifiers

```
interface-classifier {
.
.
.
rule rule-5 {
    target /sample/junose/DHCP;
    condition {
        "interfaceName=\"fastEthernet*\"";
        "interfaceName=\"atm*/*. *\"";
    }
}
```

```
    }
}
```

Classification Targets

A target is the result of the classification script that gets returned to the SAE. There are two special types of targets:

- No-match targets—Targets that begin with a - (single dash) are interpreted as no match. If the conditions of this target are matched, a no-match message is returned to SAE. You can use this type of target to exclude certain patterns or to shortcut known nonmatches. To speed up processing, use this target to specify interfaces that you do not want the SAE to manage.
- Script targets—The content of the script rule is interpreted when the classifier is initially loaded. The script rule can contain definitions of custom functions, which can be called during the matching process. Because you can insert arbitrary code into a script, you can use classification scripts to perform arbitrary tasks.

Because script targets use * (asterisks), you cannot use * in other types of targets.

Target Expressions

A target can contain expressions. These expressions can refer to an object in the SAE's memory or configuration, to specific matching conditions, or to another function or script.

Suppose the classification object in a subscriber classifier contains a field called `userName`. The classifier target `uniqueId = < - userName - >` is expanded to contain the actual content of the `userName` field before it is returned to the SAE; for example, for `userName = juser`, `uniqueId = juser` is returned.

Target expressions are enclosed in angle brackets and hyphens; for example, `< -retailerDn- >`. The classifier expands expressions before it returns the target to the SAE. The expression is interpreted by an embedded Python interpreter and can contain variables and Python operations. In the simplest case an expression can be a single variable that is replaced with its current contents. Available variable names are all fields of the object passed to the classifier and names created with regular expression matching.

Because a scripting interpreter interprets expressions, more complex operations are possible. Examples are:

- Indexing—`var[index]` returns the element index of a sequence. The first element is at index 0.
- Slicing—`var[start : end]` creates a substring of the variable `var` starting at index `start` to, but not including, index `end`; for example, `var = Hello`, `var[2:4] = ll`

Classification Conditions

You can configure multiple classification conditions for a rule. For example:

```

rule rule-2 {
  target /ent/EntDefault;
  condition {
    "pppLoginName=\"\"";
    "&interfaceName!=\"fastEthernet0*\"";
    "&interfaceName!=\"null*\"";
    "&interfaceName!=\"loopback*\"";
  }
}

```

If you prefix a condition with an & (ampersand) character, the condition is examined only if the previous condition matches.

If you prefix a condition with a | (pipe) character, the condition is examined only if the previous conditions have not produced a positive match.

You can use glob or regular expression matching to configure each target's conditions.

Glob Matching

Glob matches are of the form:

```

field = match
or
field != match

```

where match is a pattern similar to UNIX filename matching. Glob matches are case insensitive. “field != match” is true, if field = match is not true.

- *—Matches any substring.
- ?—Matches any single character.
- [range]—Matches a single character in the specified range. Ranges can have the form a-z or abcd.
- [!range]—Matches a single character outside the specified range.
- C—Matches the single character c.

The available field names are described for the specific classifiers. Examples are:

- interfaceName = fastEthernet3/0 # matches the string “fastEthernet3/0” directly.
- interfaceName = fast*3/1 # matches any string that starts with “fast” and ends with “3/1”
- interfaceName = fast*3/1.* # starts with “fast”, contains “3/1.” arbitrary ending
- interfaceName = fast*3/[2-57] # starts with “fast”, contains “3/” followed by 2,3,4,5 or 7

Regular Expression Matching

Regular expression matches are of the form:


```
field =~ re
or
field !~ re
```

where `field !~ re` is true if `field = re` is not true. The regular expression is `re`. For a complete description of the syntax, see: <http://www.python.org/doc/2.0/lib/re-syntax.html>

You can group regular expressions with pairs of parentheses. If such an expression matches, the contents of the groups are made available for target expressions. Group number *n* is available as `G[n]`, where *n* is the number of the opening parenthesis of the group. You can also name groups by using the special notation `(?P<name>...)`.

Examples:

```
ifAlias =~ "SSP(.*)"
# match a string starting with "SSP". The remainder is stored
# in the variable "G[1]"
ifAlias =~ (?P<dn>name=(?P<name>[^\,]*)).*
# match a string starting with " name=" . The whole match is
# stored in the variable " dn" . A submatch which does not
# contain any " ," -characters and starts after " name="
# is stored in variable " name"
```

- Related Topics**
- Overview of Classification Scripts on page 43
 - Classifying Interfaces (C-Web Interface)
 - Classifying Interfaces (SRC CLI) on page 49.
 - Classifying Subscribers (SRC CLI) on page 56
 - Sending DHCP Options to the JUNOS Router on page 63

Classifying Interfaces (SRC CLI)

Use the following configuration statements to define interface classification scripts:

```
shared network device name interface-classifier rule name {
    target target ;
    script script ;
    include include ;
}
shared network device 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 device.


```
user@host# edit shared network device erx-node1 interface-classifier
```
2. Create a rule for the subscriber classifier. You can create multiple rules for the classifier.

```
[edit shared network device erx-node1 interface-classifier]
user@host# edit rule rule-3
```

3. Configure either a script, a target, or an included script for the rule.

```
[edit shared network device erx-node1 interface-classifier rule rule-3]
user@host# set script script
```

OR

```
[edit shared network device erx-node1 interface-classifier rule rule-3]
user@host# set target target
```

OR

```
[edit shared sae group west-region subscriber-classifier rule rule-2]
user@host# set include include
```

Where *include* identifies a script that has already been created.

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” on page 52.

```
[edit shared network device erx-node1 interface-classifier rule rule-3]
user@host# set condition name
```

5. (Optional) Change the order of rules.

```
[edit shared network device erx-node1 interface-classifier]
user@host# insert rule rule-5 before rule-4
```

6. (Optional) Rename a rule.

```
[edit shared network device erx-node1 interface-classifier]
user@host# rename rule rule-5 to DHCP
```

7. (Optional) Verify the classifier rule configuration.

```
[edit shared network device erx-node1 interface-classifier rule rule-3]
user@host# show
target /sample/junose/PPP-special;
condition {
    "pppLoginName=\"*@special.com\"";
}
```

8. (Optional) Verify the interface classifier configuration.

```
[edit shared network device erx-node1 interface-classifier]
user@host# show
rule rule-1 {
    script "
# Use the following syntax:
#
```

```

# descr-file ::= [script] section*
# section   ::= ('[' type ']' nl conditions) | ('[*]' nl script)
# type      ::= 'a-zA-Z0-9-_*'
# nl        ::= '\\n'
# conditions ::= ((( '#' ';' ) comment) |
#                 (['&' '|' ] field-name ( '=' '|' '=' '|' != ) match) nl)*
# field-name ::= member of InterfaceObject
# match      ::= UNIX style filename matching
# script     ::= regular python script, defined functions need to be
#                 included in the list \"classify\"
#
# the section-names correspond to a PolicyList object below
# o=Policies, o=umc:
# [name] => DN: \"policyGroupName=name, o=Policies, o=umc\"
#
# Use one of the following \"field names\":
#   pppLoginName      - set to \"user@realm\", if interface is PPP
#   interfaceName     - name of the ERX Interface in CLI syntax
#   virtualRouterName - name of the VR the interface is connected to

";
}
rule rule-2 {
  script "
# apply different default policies for PPP subscribers in realm
\"special.com\"
def log(obj):
    from net.juniper.smgmt.sae import Main
    icc = Main.theComponentRegistry.get(\"icc.component\")
    if icc is None:
        Main.theComponentRegistry.put(\"icc.component\", [])
    else:
        icc.append(obj)
classify.append(log)
";
}
rule rule-3 {
  target /sample/junose/PPP-special;
  condition {
    "pppLoginName=\"*@special.com\"";
  }
}
rule rule-4 {
  target /sample/junose/PPP;
  condition {
    "pppLoginName!=\"\"";
  }
}
rule rule-5 {
  target /sample/junose/DHCP;
  condition {
    "interfaceName=\"fastEthernet*\"";
    "interfaceName=\"atm*/*.*\"";
  }
}
}

```

- Related Topics**
- Classifying Interfaces (C-Web Interface)
 - Reloading Interface Classification Scripts

- Example: Managing Specific Interfaces on page 55
- Example: Managing Interfaces by Using the Interface Description on page 55
- Overview of Configuring Classification Scripts on page 46

Interface Classification Conditions

Use the fields in this section to define interface classification conditions.

broadcastAddr

- Interface broadcast address.
- Value—Valid broadcast address format
- Example—broadcastAddr.hostAddress = “ 255.255.255.255”

ifAlias

- Description of an interface.
- Value—Interface description that is configured on the router. For JUNOSe routers, it is the description configured with the **interface description** command.
- Example—ifAlias = “ 1st pppoe int”

ifDesc

- Alternate name of the interface that is used by SNMP. This name is a system-generated name.
- Value
 - On a JUNOSe router, the format of the description is
ip<slot>/<port>.<subinterface>
 - On the JUNOS routing platform, ifDesc is the same as interfaceName.
- Example—ifDesc = “ IP3/1.1 ”

interfaceName

- Name of the interface.
- Value
 - Name of the interface in your router CLI syntax
 - FORWARDING_INTERFACE for routing instance (used by traffic mirroring)
- Example—For JUNOSe routers: interfaceName = “ fastethernet6/0.1”

For JUNOS routing platforms: interfaceName = “ fe-0/1/0.0”

For forwarding interface: interfaceName = “ FORWARDING_INTERFACE”

ipAddress

- Interface IP address.
- Value—Valid IPv4 IP address format
- Example—ipAddress = “ 10.10.30.1”

ipMask

- Interface network mask.
- Value—Valid IPv4 IP network mask format
- Example—ipMask = “ 255.255.255.255”

mtu

- Maximum transfer unit configured on the interface.
- Value—32-integer value
- Example—mtu = “ 1492”

nasPortId

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

pppLoginName

- Login name for PPP subscribers.
- Value—Login name in the format username@domain
- Example—pppLoginName = “ pebbles@virneo.net”

radiusClass

- RADIUS class attribute.
- Value—RADIUS class name
- Example—radiusClass = “ Premium”

serviceBundle

- Content of the vendor-specific RADIUS attribute for the service bundle.
- Value—Name of a service bundle

userIpAddress

- Subscriber IP address (PPP only).
- Value—valid IPv4 address
- Example—userIpAddress = “ 192.168.30.15”

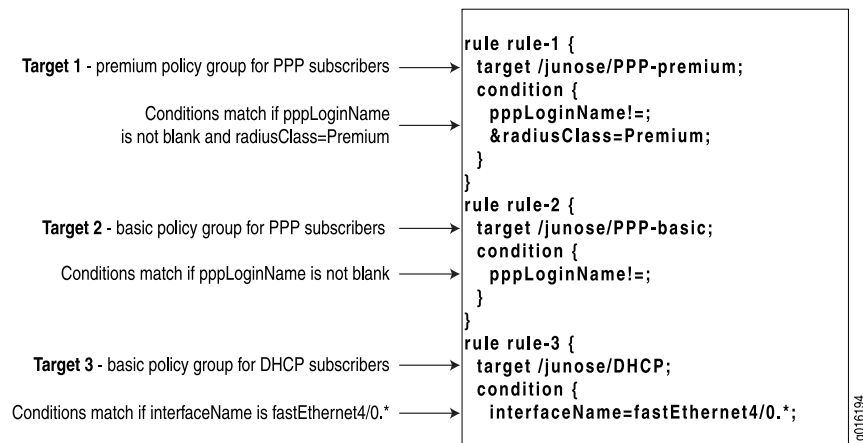
virtualRouterName

- Name of the virtual router or routing instance.
- Value—For JUNOS routers: name of the virtual router in the format vrname@hostname
For JUNOS routing platforms: name of the routing instance
- Example—virtualRouterName = “ default@erx5”

Example: Managing Interfaces for Premium and Basic PPP and DHCP Subscribers

In this scenario, the router manages two types of PPP interfaces—DHCP subscriber interfaces and static IP interfaces. The fastEthernet4/0.1 to fastEthernet4/0.999 interfaces are VLAN interfaces used to terminate DHCP subscribers.

The service provider has separated the PPP subscribers into a premium subscriber group and a basic subscriber group. These groups are distinguished by a different set of default policies applied to the PPP interface. The RADIUS class attribute in the RADIUS profile for premium subscribers is set to Premium. The rules in the interface classification script for this scenario are:



The script is processed as follows:

1. If pppLoginName is not blank and radiusClass is Premium, the PPP-premium policy group is sent to the SAE, and script processing stops.
2. If script processing proceeds and pppLoginName is not blank, the PPP-basic policy group is sent to the SAE, and script processing stops.

3. If script processing proceeds and interfaceName is fastEthernet 4/0.0 through fastEthernet 4/0.999, the DHCP policy group is sent to the SAE, and script processing stops.

- Related Topics**
- DHCP Subscriber Login and Service Activation on page 17
 - Overview of Configuring Classification Scripts on page 46
 - Overview of Classification Scripts on page 43
 - Sending DHCP Options to the JUNOS Router on page 63

Example: Managing Specific Interfaces

This example causes the SAE to load the DHCP policy group on IP interfaces on Fast Ethernet modules in slot 3/port 1, slot 1/port 1, or any port on slot 2. The SAE then manages these interfaces.

```
[edit shared network device erx-node2 interface-classifier rule rule-1]
user@host# show
target /junose/DHCP;
condition {
    interfaceName=FastEthernet3/1;
    interfaceName=FastEthernet1/1;
    interfaceName=FastEthernet2/*;
}
```

- Related Topics**
- Overview of Configuring Classification Scripts on page 46
 - Overview of Classification Scripts on page 43
 - Sending DHCP Options to the JUNOS Router on page 63
 - DHCP Options Supported on the SAE on page 74

Example: Managing Interfaces by Using the Interface Description

This example causes the SAE to load the DHCP policy group on any interface where the ifAlias starts with DHCP-subscribers.

```
[edit shared network device erx-node2 interface-classifier rule rule-2]
user@host# show
target /junose/DHCP;
condition {
    ifAlias=DHCP-subscribers*;
}
```

For this approach, you will need to use the `ip description` command to configure interface aliases that begin with DHCP-subscribers for all interfaces that support DHCP subscribers.

- Related Topics**
- Overview of Configuring Classification Scripts on page 46
 - Overview of Classification Scripts on page 43

- Sending DHCP Options to the JUNOS Router on page 63
- DHCP Options Supported on the SAE on page 74

Classifying Subscribers (SRC CLI)

Changes that you make to subscriber classification scripts do not affect subscriber sessions that are already established. One effect of this behavior is that static IP subscriber sessions are not closed if the classification script is changed in a way that would no longer cause the SAE to load a profile for certain subscribers.

On JUNOS routers that use the COPS-PR or COPS XDR router drivers, you can create a subscriber session for the router interface to start services such as script services and aggregate services. The SAE creates the router interface, but does not install any policies on it. You can create a subscriber classification rule, but not an interface classification rule for this interface.

Use the following configuration statements to define subscriber classification scripts:

```
shared sae subscriber-classifier rule name {
    target target ;
    script script ;
    include include ;
}
shared sae subscriber-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 subscriber classification scripts:

1. From configuration mode, enter the subscriber classifier configuration. In this sample procedure, the subscriber classifier is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region subscriber-classifier
```

2. Create a rule for the subscriber classifier. You can create multiple rules for the classifier.

```
[edit shared sae group west-region subscriber-classifier]
user@host# edit rule rule-2
```

3. Configure either a script, a target, or an included script for the rule.

```
[edit shared sae group west-region subscriber-classifier rule rule-2]
user@host# set target target
```

OR

```
[edit shared sae group west-region subscriber-classifier rule rule-2]
user@host# set script script
```


OR

```
[edit shared sae group west-region subscriber-classifier rule rule-2]
user@host# set include include
```

Where *include* identifies a script that has already been created.

If you configure a target, see “Subscriber Classification Targets” on page 64.

4. If you configured a target for the rule, configure a match condition for the rule. You can create multiple conditions for the rule. See “Subscriber Classification Conditions” on page 59.

```
[edit shared sae group west-region subscriber-classifier rule rule-2]
user@host# edit condition name
```

5. (Optional) Change the order of rules.

```
[edit shared sae group west-region subscriber-classifier]
user@host# insert rule rule-5 before rule-4
```

6. (Optional) Rename a rule.

```
[edit shared sae group west-region subscriber-classifier]
user@host# rename rule rule-5 to Retailer
```

7. (Optional) Verify the classifier rule configuration.

```
[edit shared sae group west-region subscriber-classifier rule rule-2]
user@host# show
target <-unauthenticatedUserDn->;
condition {
  "loginType == \"ADDR\"";
  "loginType == \"AUTHADDR\"";
}
```

8. (Optional) Verify the subscriber classifier configuration.

```
[edit shared sae group west-region subscriber-classifier]
user@host# show
rule rule-1 {
  script "# User Classification script
#
# The following attributes MAY be available for comparison.
# Attributes that are not available will have the value \"\" (empty
string).
#
# loginType: one of \"INTF\", \"AUTHINTF\", \"ADDR\", \"AUTHADDR\",
#             \"PORTAL\", \"ASSIGNEDIP\"
# userName: Everything before the \"@\" in the user's login name.
# domainName: Everything after the \"@\" in the user's login name.
# serviceBundle: A RADIUS VSA available if the login event involves
#                 authentication with a properly configured RADIUS server.
# radiusClass: The RADIUS class of user's ERX interface.
# virtualRouterName: The name of the user's virtual router.
# interfaceName: The name of the user's ERX interface (e.g.
```

```

#           \"fastEthernet3/1.0\")
#   ifAlias: The alias of the user's ERX interface, as configured on the
#   ERX.
#   ifDesc: The description of the user's ERX interface, as configured on
#           the ERX.
#   nasPortId: The user's ERX interface including Layer 2 access
#   information
#               (e.g. \"fastEthernet 3/1.0:3\")
#   macAddress: The MAC address of the user, if he is a DHCP user.
#   retailerDn: Generated by SSP for backwards compatibility; see below.
#
#   The loginType value available to this user classifier script will be
#   one of the following:
#
#   \"INTF\":
#   An INTF login is triggered every time an interface comes up and the
#   interface classifier script determines that SAE should manage that
#   interface, and the interface has not been authenticated by the router.
#
#   \"AUTHINTF\":
#   An AUTHINTF login is triggered every time an authenticated
#   interface comes up, for example as a result of an authenticated PPP
#   session.
#
#   \"ADDR\":
#   An ADDR login is triggered every time an 'unauthenticated' IP
#   address is handed out by the DHCP server in the ERX.
#
#   \"AUTHADDR\":
#   An AUTHADDR login is triggered every time an 'authenticated' IP
#   address is handed out by the DHCP server in the ERX.
#
#   \"PORTAL\":
#   A PORTAL login is triggered every time the portal API is invoked to
#   login a user.
#
#   See the customer documentation for a description of the values
#   for each login type available in the script.
#
#   One of the values available during some types of logins is the
#   'retailerDn'. This is a generated value available for backwards
#   compatibility with previous versions of SAE. SAE generates this
#   value as follows:
#
#   The retailerDn value is generated by, first, determining an
#   effective user domain name, and second, locating the retailer
#   entry in LDAP that contains that effective domain name. If no
#   such retailer exists, the retailerDn value will be \"\".
#
#   The effective user domain name is the first of the following that yields
#   a result:
#
#   1. For PPP, PORTAL, and PUBLIC logins where a non-empty domainName
#      is supplied, that non-empty domain name is used as the effective
#      domain name.
#
#   2. For INTF logins, and for PPP, PORTAL, and PUBLIC logins where a
#      non-empty domain name is not supplied, the effective domain name
#      is the name of the user's virtual router, unless that effective
#      domain does not exist in some retailer in LDAP.
#

```

```

# 3. If neither step 1 nor step 2 yields an effective domain name,
#   \"default\" is used as the effective domain name.
#

";
}
rule rule-2 {
  target <-unauthenticatedUserDn->;
  condition {
    "loginType == \"ADDR\"";
    "loginType == \"AUTHADDR\"";
  }
}
rule rule-3 {
  target <-retailerDn->??sub?(uniqueID=<-userName->);
  condition {
    "retailerDn != \"\"";
    "& userName != \"\"";
  }
}
}

```

- Related Topics**
- Classifying Subscribers (C-Web Interface)
 - Sending DHCP Options to the JUNOS Router on page 63
 - Example: Subscriber Classification Scripts for Static IP Subscriber on page 65
 - Overview of Classification Scripts on page 43
 - Overview of Subscribers on page 3

Subscriber Classification Conditions

Subscriber classification conditions define match criteria that are used to find the subscriber profile. Use the fields in this section to define subscriber classification conditions.

dhcp

- DHCP options. See “Sending DHCP Options to the JUNOS Router” on page 63.

domainName

- Domain name of the subscriber.
- Value—Valid domain name
- Example—domainName = “isp99.com”

ifAlias

- Description of the interface.
- Value—Interface description that is configured on the router. For JUNOSe routers, it is the description configured with the **interface description** command
- Example—ifAlias = “ dhcp-subscriber12”

ifDesc

- Alternate name for the interface that is used by SNMP. This name is a system-generated name.
- Value
 - On a JUNOSe router, the format of the description is
ip<slot>/<port>.<subinterface>
 - On the JUNOS routing platform, ifDesc is the same as interfaceName.
- Example—ifDesc = “ IP3/1.1 ”

interfaceName

- Name of the interface.
- Value
 - Name of the interface in your router CLI syntax
 - FORWARDING_INTERFACE for routing instance (used by traffic mirroring)
 - Router for a JUNOSe router instance
- Example—For JUNOSe routers: interfaceName = “ fastEthernet6/0”

For JUNOS routing platforms: interfaceName = “ fe-0/1/0.0”

For forwarding interface: interfaceName = “ FORWARDING_INTERFACE”

loginName

- Name to be used to create a loginName attribute for a subscriber session for JUNOS interfaces that are not otherwise assigned a loginName when a session starts, such as unauthenticated DHCP addresses, unauthenticated IP interfaces (that are not using PPP connections), or core-facing interfaces.

The loginName can also be used to identify a subscriber session through the SAE CORBA remote API.

- Value—Name in the form subscriber@domain
- Guideline—The format is not defined. A loginName can be of form subscriber, domain\subscriber, subscriber@domain, or as otherwise defined by the login setup of the operator.
- < Login name >
- Example—idp@idp

loginType

- Type of subscriber session to be created.
- Value—One of the following login types:
 - ASSIGNEDIP—For assigned IP subscribers. Triggered when an application accesses a subscriber object for an assigned IP subscriber that is not currently loaded into memory. (Supported on JUNOS routers.)
 - AUTHINTF—For authenticated interface login requests. Triggered when a login Name is reported together with the interface, such as authenticated PPP or autoconfigured ATM interface, by means of the **subscriber** command. (Supported on JUNOS routers.)
 - INTF—For unauthenticated interface login requests. Triggered when an interface comes up and the interface classification script determines that the SAE should manage the interface. (Supported on JUNOS routing platforms and JUNOS routers.)
 - ADDR—For unauthenticated address login requests. Triggered when the DHCP server in the JUNOS router provides an unauthenticated IP address. (Supported on JUNOS routers.)
 - AUTHADDR—For authenticated address login requests. Triggered when the DHCP server in the JUNOS router provides an authenticated IP address. (Supported on JUNOS routers.)
 - PORTAL—Triggered when the portal API is invoked to log in a subscriber. (Supported on JUNOS routing platforms and JUNOS routers.)
- Example—loginType = “ AUTHADDR”

macAddress

- String representation of the DHCP subscriber media access control (MAC) address.
- Value—Valid MAC address
- Example—macAddress = “ 00:11:22:33:44:55”

nasPortId

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

radiusClass

- RADIUS class used for authorization.
- Value—RADIUS class name
- Example—radiusClass = “ Premium”

retailerDn

- DN of the retailer object. The object is found when the domain name is mapped to a retailer object in LDAP.
- Value—DN of a retailer

serviceBundle

- Content of the vendor-specific RADIUS attribute for the service bundle.
- Value—Name of a service bundle
- Example—serviceBundle = “ goldSubscriber”

unauthenticatedUserDn

- DN of the unauthenticated subscriber profile (usable for target expressions only).
- Value—DN of a subscriber profile

userName

- Name of the subscriber.
- Value—Subscriber name without the domain name
- Example—userName = “ peter”

virtualRouterName

- Name of the virtual router or routing instance.
- Value—For JUNOS routers: name of the virtual router in the format `vrname@hostname`

For JUNOS routing platforms: name of the routing instance
- Example—`virtualRouterName = " default@e_series5"`

Sending DHCP Options to the JUNOS Router

Subscriber classification scripts support DHCP options conveyed through COPS. When COPS reports an address, the JUNOS router sends DHCP options received for DHCP requests for that address. The DHCP options are available in the subscriber classification context for selecting the subscriber profile to load.

The fields in Table 7 on page 63 are in the classification context of subscriber classification scripts.

Table 7: DHCP Options in UserClassificationContext Field

DHCP Option	UserClassificationContext Field	Comments
giAddr	dhcp.giAddr	Relay agent gateway address
Option 82 data	dhcp.getOption(82)	Content is accessible with <code>getSubOptions()</code>
Client ID	dhcp.getOption(61).getString()	
Lease time	dhcp.getOption(51).getInt()	
Client requested parameter list	dhcp.getOption(55).getBytes()	
Domain name sent to client	dhcp.getOption(12).getString() dhcp.getOption(15).getString()	12 = HostName 15 = DomainName
DNS server address(es) sent to client	dhcp.getOption(6).getIpAddresses()	
Subnet mask	dhcp.getOption(1).getIpAddress()	
NetBios name server address(es) sent to client	dhcp.getOption(44).getIpAddresses()	
NetBios node type	dhcp.getOption(46).getBytes()	
Default router address(es) sent to client	dhcp.getOption(3).getIpAddresses()	

The DHCP options are accessible to the subscriber classification script with the following syntax:

```
dhcp.giAddr = " match"

# interpret option 61 as string
dhcp[61].string = " match"

# interpret option 1 (subnet) as dotted decimal IP
dhcp[1].ipAddress = " match"

# option 82, suboption 1, interpreted as string
dhcp[82].subOptions[1].string = " match"
```

The received DHCP options are also stored in the UserSession and are available through the portal API (method User.getDhcpOptions).

- Related Topics**
- Selecting DHCP Parameters on page 73
 - Classifying DHCP Subscribers (SRC CLI) on page 68
 - Creating DHCP Profiles (C-Web Interface)
 - DHCP Options Supported on the SAE on page 74

Subscriber Classification Targets

The target of the subscriber classification script is an LDAP search string. The search string uses a syntax similar to an LDAP URL (see *RFC 2255—The LDAP URL Format (December 1997)*).

The syntax is:

```
" baseDN [ ? [ attributes ] [ ? [ scope ] [ ? [ filter ] ] ] ]"
```

- baseDN—Distinguished name of object where the LDAP search starts
- attributes—Can be used to override attributes in the loaded LDAP object. For example, for static IP subscribers the SAE must learn the IP address assigned to a particular subscriber. This address is defined in the ipAddress attribute of the subscriber profile. A target of the form
baseDN?ipAddress = <-function(interfaceName)-> invokes function after the subscriber profile is loaded from LDAP and sets the IP address to the return value of function. The function is defined in the subscriber classification script, and can be used for a variety of things; for example, to query an external database.



NOTE: You can use subscriber classification to override only the ipAddress, loginName, or accountingId attributes. If you configure values to override other attributes, the value is lost when the SAE recovers from a network or server failure.

- scope—Scope of search

- base—Is the default, searches the base DN only.
- one—Searches the direct children of the base DN.
- sub—Searches the complete subtree below the base DN.
- filter—Is an RFC 2254–style LDAP search filter expression; for example, (uniqueId = <-userName->). See *RFC 2254—The String Representation of LDAP Search Filters (December 1997)*.

With the exception of baseDN all the fields are optional.

The result of the LDAP search must be exactly one directory object. If no object or more than one object is found, the subscriber session is terminated.

Related Topics

- Overview of Classification Scripts on page 43
- Overview of Configuring Classification Scripts on page 46
- Classifying Subscribers (SRC CLI) on page 56
- Syntax for DHCP Classification Targets on page 72
- Classifying DHCP Subscribers (C-Web Interface)

Example: Subscriber Classification Scripts for Static IP Subscriber

In cases such as bridged 1483 DSL with a single subscriber, you can write the subscriber classification script so that it loads a specific subscriber profile. If the interface is matched to a subscriber profile, a subscriber session is immediately established. An SAE application (for example, a portal) can still force the subscriber with this subscriber profile to perform a Web login.

One way to achieve the mapping of subscriber interface to subscriber profile is to provision the assigned interface name in the associated subscriber profile in LDAP. In this case the subscriber classification script can include a rule like this:

```
[edit shared sae group west-region subscriber-classifier rule rule-1]
user@host# show
target retailerName=default,o=Users,o=umc??sub?(interfaceName=<-interfaceName->);
condition {
    "loginType=="INTF\"";
    " &interfaceName=fastEthernet*" ;
}
```

Another way may include a special encoding of the interface alias (ifAlias) field of the subscriber interface. This encoding must then be provisioned when the interface for the subscriber is provisioned. In this example, the encoding SAE-username is chosen for ifAlias; for example, for subscriber juser the interface alias would be set to SAE-juser. The match is performed with a regular expression, which separates the user ID from the ifAlias prefix.

```
[edit shared sae group west-region subscriber-classifier rule rule-1]
user@host# show
```

```
target retailerName=default,o=Users,o=umc??sub?(uniqueID=<-userId>);
condition {
    "loginType=="INTF\"";
    " &ifAlias=~SAE-(?P<userId>.*)" ;
}
```

- Related Topics**
- Overview of Configuring Classification Scripts on page 46
 - Overview of Classification Scripts on page 43
 - Static IP Subscribers on page 23
 - Overview of Subscribers on page 3

Example: Subscriber Classification Scripts Using a Subscriber Group

To support scenarios in which the SAE has no access to the subscriber database, the SAE can load anonymous profiles for groups of subscribers. The following example loads a particular subscriber profile when subscribers of domain another-isp.com log in

```
[edit shared sae group west-region subscriber-classifier rule rule-1]
user@host# show
target uniqueID=anon,ou=default,retailerName=another-isp,o=Users,o=umc;
condition {
    " domainName=another-isp.com" ;
}
```

- Related Topics**
- Overview of Configuring Classification Scripts on page 46
 - Overview of Classification Scripts on page 43
 - Overview of Subscribers on page 3

Example: Subscriber Classification Scripts for Enterprise Subscribers

For enterprise subscribers, you can create one general subscriber classifier script that matches a unique subscriber profile to each managed router interface. The subscriber profile is the access subscription that represents an Internet access in an enterprise. The following examples show two approaches to creating the general classifier script. You can use one of these strategies or a combination of strategies.

Matching on the Interface Name

In this scenario, you configure the interface name field in the access subscription for the site to match an interface on the router. The format for the interface name could be: interfaceName@virtualRouterName@routerName. You then create a classification script that searches for subscriber profiles that match a specific interface. For example:

```
[edit shared sae group west-region subscriber-classifier rule rule-1]
user@host# show
target ou=Managed
CPE,retailerName=Retailer-Two,o=Users,o=UMC??sub?(interfaceName=<-interfaceName->@<-virtualRouterName->);
```

```
condition {
  "loginType=="INTF\"";
  &interfaceName=="fe*\\" " " ;
}
```

Matching on the Interface Alias

For JUNOS routers, you can configure the interface description on the router in a format that the classifier script can match to the interface alias in an access subscription. In a simple case, you can configure the interface description only for interfaces that terminate a managed CPE, and match them to the interface alias in the directory. The subscriber classifier could be configured as follows:

```
[edit shared sae group west-region subscriber-classifier rule rule-1]
user@host# show
target ou=Managed CPE,retailerName=Retailer-Two,o=Users,o=UMC??sub?(interfaceAlias=<-ifAlias->);
condition {
  ifAlias != \\"\"
}
```

- Related Topics**
- Overview of Configuring Classification Scripts on page 46
 - Overview of Classification Scripts on page 43
 - Overview of Subscribers on page 3
 - Enterprise Subscriber Login Process on page 26

Example: Creating Router Interface Subscriber Session

Aggregate services or script services can be activated on a router instead of an interface or DHCP address. On JUNOS routers that use the COPS-PR or COPS XDR router driver, the SAE automatically creates a router interface; and then a subscriber session as specified by the subscriber classification script.

For example, the following script searches for a router profile in the directory under `ou = routers`, `retailerName = default`, `o = Users`, `o = umc`, with a `routerName` attribute that matches the virtual router name (such as `default@erx-node1`).

```
[edit shared sae group west-region subscriber-classifier rule rule-1]
user@host# show
target ou=routers,retailername=default,o=Users,o=UMC??sub?(routerName=<-virtualRouterName->);
condition {
  "interfaceName=="Router\"";
}
```

- Related Topics**
- Overview of Classification Scripts on page 43
 - Overview of Configuring Classification Scripts on page 46
 - Sending DHCP Options to the JUNOS Router on page 63
 - Assigning DHCP Addresses to Subscribers on page 84

Example: Activating Services for a Group of Subscriber Sessions

A subscriber classification script can assign a shared subscriber profile and a login name to a subscriber session for a group of interface subscriber sessions. The following example assigns the login name `idp@idp` to subscriber sessions for JUNOS interfaces that have core specified as the `ifAlias` (as configured on the JUNOS router).

```
[edit shared sae group IDP subscriber-classifier rule rule-3]
root@buffy# show
target routerName=idp,ou=interfaces,retailname=SP-IDP,o=Users,o=UMC?loginName=idp@idp;
condition {
  "ifAlias=="core\"";
}
```

You can use this type of subscriber classification to activate a service for a group of interface subscriber sessions that are to be treated the same. For example, in the configuration for an aggregate service, a fragment service could be created for all subscriber interface sessions on interfaces identified by the `ifAlias` core on a virtual router. The subscriber reference expression in the configuration for the fragment service would reference the virtual router name and the login name, such as `vr = "<- virtualRouterName ->", login_name = "idp@idp."`

You can also use the SAE CORBA remote API to get lists of the subscriber sessions that share the same login name.

- Related Topics**
- Overview of Classification Scripts on page 43
 - Overview of Configuring Classification Scripts on page 46
 - Connections to Managed Devices

Classifying DHCP Subscribers (SRC CLI)

Use the following configuration statements to configure DHCP classification scripts:

```
shared sae dhcp-classifier rule name {
  target target ;
  script script ;
  include include ;
}
shared sae dhcp-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 configure DHCP classification scripts:

1. From configuration mode, enter the DHCP classifier configuration. In this sample procedure, the classifier is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region dhcp-classifier
```

2. Create a rule for the subscriber classifier. You can create multiple rules for the classifier.

```
[edit shared sae group west-region dhcp-classifier]
user@host# edit rule rule-1
```

3. Configure either a target or a script for the rule.
4. (Optional) Configure a target, script, or included script for the rule.

```
[edit shared sae group east-region dhcp-classifier rule rule-1]
user@host# set target target
```

OR

```
[edit shared sae group east-region dhcp-classifier rule rule-1]
user@host# set script script
```

OR

```
[edit shared sae group west-region subscriber-classifier rule rule-2]
user@host# set include include
```

Where *include* identifies a script that has already been created.

If you configure a target, see “Syntax for DHCP Classification Targets” on page 72.

5. If you configured a target for the rule, configure a match condition for the rule. You can create multiple conditions for the rule. See “DHCP Classification Conditions” on page 70.

```
[edit shared sae group east-region dhcp-classifier rule rule-1]
user@host# edit condition name
```

6. (Optional) Change the order of rules.

```
[edit shared sae group east-region dhcp-classifier]
user@host# insert rule rule-5 before rule-4
```

7. (Optional) Rename a rule.

```
[edit shared sae group east-region dhcp-classifier]
user@host# rename rule rule-2 to dhcp
```

8. (Optional) Verify the classifier rule configuration.

```
[edit shared sae group east-region dhcp-classifier rule rule-1]
user@host# show
target cn=default,<-dhcpProfileDN->;
condition {
  1;
}
```

9. (Optional) Verify the DHCP classifier configuration.

```
[edit shared sae group west-region dhcp-classifier]
user@host# show
rule rule-1 {
  script "# DHCP classification script
#
# The DHCP classification script can use the following fields:
#
# interfaceName      - interface where DHCP DISCOVER was received.
# ifAlias            - \"ip description\" of interface
# ifDesc             - SNMP standard name of interface
# nasPortId
# virtualRouterName  - VR where DHCP DISCOVER was received
# macAddress         - MAC address of DHCP client
# dhcp               - DHCP options
# poolName           - DHCP Pool name set by authorization plug-in
# authVirtualRouterName - VR name set by authorization plug-in
# dhcpProfileDN      - search base for DHCP Profiles

";
}
rule rule-2 {
  target cn=default,<-dhcpProfileDN->;
  condition {
    1;
  }
}
```

- Related Topics**
- Sending DHCP Options to the JUNOS Router on page 63
 - Selecting DHCP Parameters on page 73
 - Creating DHCP Profiles (SRC CLI) on page 77
 - Classifying DHCP Subscribers (C-Web Interface)
 - DHCP Options Supported on the SAE on page 74

DHCP Classification Conditions

DHCP classification conditions define match criteria that are used to find the DHCP profile. Use the fields in this section to define DHCP classification conditions.

authVirtualRouterName

- Name of JUNOS virtual router that is set by an authorization plug-in through the authorization response.
- Value—Name of the virtual router in the format `vrname@hostname`

dhcp

- DHCP options. See “DHCP Options Supported on the SAE” on page 74 .

dhcpProfileDN

- Search base for DHCP profiles. The DN can be used in target expressions.
- Value—DN of DHCP profile

interfaceName

- Name of the interface where the DHCP discover message was received.
- Value—Name of the interface in your router CLI syntax
- Example—interfaceName = fastEthernet6/0

ifAlias

- Description of the interface where the DHCP discover request was received.
- Value—Interface description that is configured on the router. For JUNOS routers, it is the description configured with the **interface description** command
- Example—ifAlias = “ dhcp-subscriber12”

ifDesc

- Alternate name for the interface where the DHCP discover request was received. This is a system-generated name that is used by SNMP.
- Value
 - On a JUNOS router, the format of the description is:
ip<slot>/<port>.<subinterface>
 - On the JUNOS routing platform, ifDesc is the same as interfaceName.

macAddress

- MAC address of the DHCP client that appears in DHCP request.
- Value—Valid MAC address
- Example—macAddress = “ 00:11:22:33:44:55”

nasPortId

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

poolName

- IP address pool name that is set by an authorization plug-in through the authorization response.
- Value—Name of an address pool configured on the JUNOS router

virtualRouterName

- Name of the virtual router.
- Value—Name of the virtual router in the format `vrname@hostname`

Syntax for DHCP Classification Targets

The target of the DHCP classification script uses a syntax similar to an LDAP URL. With the exception of baseDN, all fields are optional. The syntax is:

```
baseDN [ ? [ attributes ] [ ? [ scope ] [ ? [ filter ] ] ] ]
```

- baseDN—DN of object where search starts.
- attributes—Comma-separated list of properties, in the format `attribute = <-value->`, that allow you to set specific attributes for directory objects that the script finds; see “DHCP Classification Conditions” on page 70.

You can use the attribute configuration to override attributes in the directory. For example, to override the IP pool name that is stored in the DHCP profile with the pool name that the authorization plug-in sends, use the attribute statement `radiusFramedPool = <-poolName->`.

- scope—Scope of search in the directory
 - base—Searches the base DN only; default scope
 - one—Searches the direct subordinates of the base DN (one-level search)
 - sub—Searches all objects subordinate to the base DN
- filter—An RFC 2254-style LDAP search filter expression; for example, `(uniqueId = <-userName->)`. See *RFC 2254—The String Representation of LDAP Search Filters (December 1997)*.

Related Topics

- Selecting DHCP Parameters on page 73
- Classifying DHCP Subscribers (SRC CLI) on page 68
- Creating DHCP Profiles (SRC CLI) on page 77
- Subscriber Classification Targets on page 64
- Creating DHCP Profiles (C-Web Interface)
- DHCP Options Supported on the SAE on page 74

Selecting DHCP Parameters

The SAE sends a set of parameters to the DHCP server in the JUNOSe router. The DHCP server determines the IP address offered, as well as the options sent to the DHCP client. The parameters comprise IP address authorization parameters, as well as parameters stored in a DHCP profile. Parameters in the DHCP profile override authorization parameters.



NOTE: JUNOSe routers do not currently support the functionality described in this section. DHCP options and BOOTP options that the SAE sends to the JUNOSe router are ignored.

DHCP servers use DHCP options to configure DHCP clients. The DHCP local server in the JUNOSe router supports a subset of DHCP options. The SAE supports all DHCP options defined in *RFC 2132—DHCP Options and BOOTP Vendor Extensions (March 1997)* by name. It also supports other options, but you need to specify them by number and type. The DHCP options allow a flexible definition of parameters offered to DHCP subscribers. For example, they allow integration with cable modems or set-top boxes because you can configure options to control the boot sequence of these devices.

You can configure DHCP options in DHCP profiles and in DHCP classification scripts. Table 8 on page 74 lists the name, number, and type of all supported DHCP options. You can use these fields to configure DHCP options.

The following example shows how to specify an option by number and by type. The two statements identify the same option:

```
dhcp[12]

dhcp['host-name']
```

In SDX software earlier than Release 4.2, you had to include the option type in your option definition. For example:

```
dhcp[12].string = HOST
```

You can now write:

```
dhcp[12] = HOST
```

Note that the earlier method of defining options still works in Release 4.2 and later.

- Related Topics**
- Assigning DHCP Addresses to Subscribers on page 84
 - DHCP Subscriber Login and Service Activation on page 17
 - DHCP Options Supported on the SAE on page 74
 - Creating DHCP Profiles (SRC CLI) on page 77

DHCP Options Supported on the SAE

Table 8 on page 74 lists the DHCP options are available.

Table 8: DHCP Options Supported on the SAE

Option Name	Option Number	Option Type
subnet-mask	1	ip-address
time-offset	2	int32
routers	3	ip-address
time-servers	4	ip-address
ien116-name-servers	5	ip-address
domain-name-servers	6	ip-address
log-servers	7	ip-address
cookie-servers	8	ip-address
lpr-servers	9	ip-address
impress-servers	10	ip-address
resource-location-servers	11	ip-address
host-name	12	string
boot-size	13	int16
merit-dump	14	string
domain-name	15	string
swap-server	16	ip-address
root-path	17	string
extension-path	18	string
ip-forwarding	19	int8
non-local-source-routing	20	int8
policy-filter	21	ip-address
max-dgram-reassembly	22	int16
default-ip-ttl	23	int8
path-mtu-aging-timeout	24	int32
path-mtu-plateau-table	25	int16

Table 8: DHCP Options Supported on the SAE *(continued)*

Option Name	Option Number	Option Type
interface-mtu	26	int16
all-subnets-local	27	int8
broadcast-address	28	ip-address
perform-mask-discovery	29	int8
mask-supplier	30	int8
router-discovery	31	int8
router-solicitation-address	32	ip-address
static-routes	33	ip-address
trailer-encapsulation	34	int8
arp-cache-timeout	35	int32
ieee802-3-encapsulation	36	int8
default-tcp-ttl	37	int8
tcp-keepalive-interval	38	int32
tcp-keepalive-garbage	39	int8
nis-domain	40	string
nis-servers	41	ip-address
ntp-servers	42	ip-address
netbios-name-servers	44	ip-address
netbios-dd-server	45	ip-address
netbios-node-type	46	int8
netbios-scope	47	string
font-servers	48	ip-address
x-display-manager	49	ip-address
requested-ip-address	50	ip-address
ip-address-lease-time	51	int32
option-overload	52	int8
dhcp-msg-type	53	int8

Table 8: DHCP Options Supported on the SAE *(continued)*

Option Name	Option Number	Option Type
server-identifier	54	ip-address
parameter-request-list	55	data-string
message	56	string
maximum-dhcp-msg-size	57	int16
renewal-time	58	int32
rebinding-time	59	int32
vendor-class-identifier	60	data-string
client-identifier	61	data-string
nisplus-domain	64	string
nisplus-servers	65	ip-address
tftp-server-name	66	string
bootfile-name	67	string
mobile-ip-home-agent	68	ip-address
smtp-server	69	ip-address
pop-server	70	ip-address
nnntp-server	71	ip-address
www-server	72	ip-address
finger-server	73	ip-address
irc-server	74	ip-address
streettalk-server	75	ip-address
streettalk-directory-assistance-server	76	ip-address

Related Topics

- Selecting DHCP Parameters on page 73
- Classifying DHCP Subscribers (SRC CLI) on page 68
- Creating DHCP Profiles (SRC CLI) on page 77
- Sending DHCP Options to the JUNOS Router on page 63
- Syntax for DHCP Classification Targets on page 72
- DHCP Classification Conditions on page 70

Creating DHCP Profiles (SRC CLI)

When the SAE receives a DHCP discover request from the router, it uses the client's MAC address to find a DHCP profile in cache or in the directory. If it finds a DHCP profile, the SAE uses the information in the profile to create a discover decision that it returns to the router. The discover decision includes information to select an IP address and DHCP options to configure the DHCP client.

When a DHCP subscriber logs in to the SAE through a Web portal, the SAE registers the subscriber's equipment and creates a cached DHCP profile in the *o = AuthCache* directory. These profiles are keyed by the MAC address of the DHCP client device. They are created by the `grantPublicIp` or the `registerEquipment` methods.

DHCP profiles are stored in the *o = AuthCache* directory in the `dhcpProfile` object class. The `dhcpProfile` object class is subordinate to the `cachedAuthenticationProfiles` object class. Manually created profiles are keyed by the `cn` (common name) attribute.

For more information about how the SAE handles DHCP subscribers, see:

- Assigning DHCP Addresses to Subscribers on page 84
- DHCP Subscriber Login and Service Activation on page 17

Use the following configuration statements to create a DHCP profile:

```
shared auth-cache cached-dhcp-profile name {
  description description ;
  pool-name pool-name ;
  ip-address ip-address ;
  dhcp-options dhcp-options ;
  boot-server-name boot-server-name ;
  boot-file-name boot-file-name ;
  virtual-router virtual-router ;
  local-interface local-interface ;
  lease-time lease-time ;
  user-name user-name ;
  service-bundle service-bundle ;
  radius-class radius-class ;
}
```

To create a DHCP profile:

1. From configuration mode, enter the DHCP cached authentication profile configuration.

```
user@host# edit shared auth-cache cached-dhcp-profile default
```

2. (Optional) Configure a description for the profile.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set description description
```

3. (Optional) Configure the name of the IP address pool on the JUNOS router from which a DHCP address is selected.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set pool-name pool-name
```

4. (Optional) Configure the fixed IP address that is offered to the DHCP client if the client is part of a network in the configured DHCP pool.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set ip-address ip-address
```

5. (Optional) Configure the DHCP options that are used to configure DHCP clients.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set dhcp-options dhcp-options
```

6. (Optional) Configure the name of the server used to boot the DHCP client.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set boot-server-name boot-server-name
```

7. (Optional) Configure the name of a boot file used to boot the DHCP client.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set boot-file-name boot-file-name
```

8. (Optional) Configure the name of the JUNOS virtual router that holds the IP address pool.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set virtual-router virtual-router
```

9. (Optional) Configure the name of the JUNOS interface that is used to check the validity of system-created DHCP profiles.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set local-interface local-interface
```

10. (Optional) Configure the length of time the supplied IP address is valid.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set lease-time lease-time
```

11. (Optional) Configure the name of DHCP user without the domain name.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set user-name user-name
```

12. (Optional) Configure the vendor-specific RADIUS attribute that specifies the SRC service bundle to use.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set service-bundle service-bundle
```

13. (Optional) Configure the RADIUS attribute class.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# set radius-class radius-class
```

14. (Optional) Verify your configuration.

```
[edit shared auth-cache cached-dhcp-profile default]
user@host# show
description "This DHCP profile is used to select addresses from the
\"default\"
pool.";
virtual-router *;
local-interface *;
```

- Related Topics**
- Selecting DHCP Parameters on page 73
 - Classifying DHCP Subscribers (SRC CLI) on page 68
 - Syntax for DHCP Classification Targets on page 72
 - DHCP Options Supported on the SAE on page 74
 - DHCP Classification Conditions on page 70

Chapter 5

Overview of Plug-Ins Included with the SAE

- How Internal Plug-Ins Work on page 81
- Types of Internal Plug-Ins on page 82
- Assigning DHCP Addresses to Subscribers on page 84
- Creating and Tracking Subscriber Sessions on page 86
- Activating and Tracking Service Sessions on page 87

How Internal Plug-Ins Work

Plug-ins work with the SAE through events. Events such as subscriber logins and logouts, as well as service activation and deactivation, trigger the SAE to create event objects and send them to plug-in instances that are configured to receive the events. When a plug-in receives an event, it processes the event. For example, when a subscriber logs in, the SAE sends the username and password to an authentication plug-in that compares the username and password with data stored in a directory.

The plug-in configuration is made up of a plug-in pool and event publishers.

Plug-In Pool

The plug-in pool consists of plug-in instances. A plug-in instance describes a particular plug-in that can handle events that it receives from the SAE. An authorization plug-in instance might be set up to perform RADIUS authentication when it receives a subscriber login event. A tracking plug-in instance might be set up to write accounting information to a file when it receives service session events.

For each type of plug-in you can create multiple instances that contain different configurations of the plug-in.

If you have multiple retailers, you might use different authentication methods and servers to authenticate each retailer's subscribers. In this case you could set up an authentication plug-in instance for each retailer.

You could also set up a tracking plug-in instance to write certain accounting information to a file whenever it receives an event. Then you could set up another instance that writes different accounting information to a different file. You could

then use one instance to track subscriber sessions and another to track service sessions. Or you could set up plug-in instances to track different types of services.

Event Publishers

Event publishers tell the SAE which events to send to which plug-in instances. There are four types of event publishers. Each type determines the scope of events that are sent to plug-in instances.

- Service-specific publishers—Authenticate subscribers of a particular service, authorize sessions for the service, and track subscriber activity related to the service
- Retailer-specific publishers—Authenticate and track subscribers and authorize DHCP address allocations for subscribers who log in to the domain(s) of a particular retailer
- Virtual router-specific publishers—Authenticate and track managed interfaces on a particular virtual router
- Global publishers—Authorize all subscriber sessions, track all subscriber and service sessions, authorize DHCP address allocations for all DHCP subscribers, and authorize all subscribers to change their subscriptions; authenticate subscribers and authorize DHCP address allocations for subscribers who log in to a retailer domain for which no retailer-specific authentication plug-ins are specified; and track all router interfaces that the SAE manages

Each publisher can notify a number of plug-in instances when an event occurs, and each plug-in instance can be registered with a number of publishers.

Related Topics

- Types of Internal Plug-Ins on page 82
- SAE Plug-Ins
- SAE Accounting
- Configuring Internal Plug-Ins on page 89
- Configuring the SAE for External Plug-Ins on page 90

Types of Internal Plug-Ins

There are two main types of plug-ins: authorization plug-ins and tracking plug-ins.

Authorization Plug-Ins

Authorization plug-ins can perform both authentication (that is, verify the originator of a request) and authorization. Authorization can include the setting of service session parameters such as session timeout or authorizing services based on the current load of the router.

You can set up authorization plug-ins to:

- Globally authorize all subscriber sessions.
- Authenticate subscribers who belong to a particular retailer's domain.
- Globally authenticate and/or authorize all service sessions.
- Authenticate and/or authorize sessions for a particular service.
- Globally authorize DHCP address allocations.
- Authorize DHCP address allocation for subscribers who log in to a particular retailer's domain.
- Globally authorize subscribers to change their subscriptions.



NOTE: Event publishers send events to all configured plug-in instances. For authentication to succeed, all authentication plug-ins that receive the authentication request must grant authentication.

Tracking Plug-Ins

Tracking plug-ins track activity or log accounting information. You can set up tracking plug-ins to:

- Globally track all subscribers.
- Track subscribers who belong to a particular retailer's domain.
- Globally track all service sessions.
- Track service sessions for individual services.
- Track QoS service sessions for individual services and attach the required QoS profile to the JUNOS subscriber interface.

Tracking plug-ins keep the state of active sessions and provide usage and accounting data. For each subscriber and service session, plug-ins can track when the session is activated and deactivated and can keep interim updates. For example, when the SAE activates a service, it sends a Service Session Start event to tracking plug-in instances that are registered to receive events for that service. When the service is stopped, the SAE sends a Service Session Stop event to all tracking plug-ins that received the Service Session Start event. If interim accounting is configured, service session interim update events are sent at regular intervals to all tracking plug-ins that are registered to receive the event.

One application of tracking plug-ins is to keep usage records, such as session time and volume counters. Service-tracking plug-ins can set a timeout for a service session in response to start and interim updates that the plug-in receives for the session. When a service session is active longer than the defined timeout, the SAE stops the session and sends service session stop events to the tracking plug-ins.

Another application is to track QoS services and attach the required QoS profile to the subscriber interface. See Overview of QoS on JUNOS Routers.

Customizing RADIUS Packets with Plug-Ins

RADIUS internal plug-ins include flexible RADIUS plug-ins and custom RADIUS plug-ins that let you customize RADIUS authentication and accounting packets that the SAE sends to RADIUS servers. You can specify which fields are included in various types of RADIUS packets and what information is contained in the fields.

For example, you can specify values in authentication response packets that will set session and idle timeouts, set the RADIUS class, and set the session volume quota. For accounting packets, you can specify which fields to include in accounting records.

For DHCP subscribers, you can set up RADIUS authorization plug-ins to return to the router attributes that can be used to select a DHCP address or select a fixed address for each subscriber.

The main difference between flexible RADIUS plug-ins and custom RADIUS plug-ins is that custom plug-ins are designed to deliver better system performance than the flexible RADIUS plug-ins. To use a custom plug-in, you must provide a Java class that implements the SPI defined in the RADIUS client library. Use this SPI to specify which fields and field values to include in RADIUS accounting packets. The RADIUS client library is part of the SAE core API.

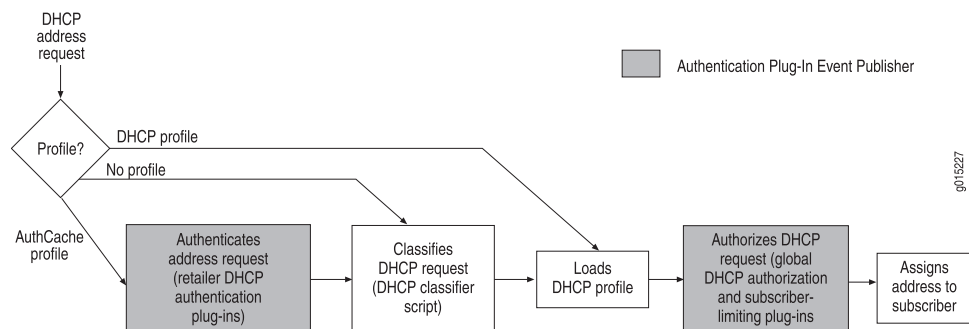
To customize RADIUS packets with a flexible RADIUS plug-in, see “Overview of Flexible RADIUS Plug-Ins” on page 118.

- Related Topics**
- How Internal Plug-Ins Work on page 81
 - Configuring Internal Plug-Ins on page 89
 - Creating and Tracking Subscriber Sessions on page 86
 - Configuring Tracking Plug-Ins on page 96
 - Configuring the SAE for External Plug-Ins on page 90

Assigning DHCP Addresses to Subscribers

Figure 18 on page 84 shows the process that the SAE uses to assign addresses to DHCP subscribers.

Figure 18: DHCP Address Assignment



To create and track a subscriber session for DHCP subscribers, the SAE:

1. Uses the client's media access control (MAC) address to look up a profile in cache or in the directory.
 - a. If the SAE finds an authCache profile, it skips to authenticating the address request.
 - b. If the SAE does not find a profile, it skips to classifying the DHCP request.
 - c. If the SAE finds a DHCP profile, it skips to loading a DHCP profile.

2. Authenticates the address request.

The SAE authenticates the request by using the configured DHCP authentication plug-ins. The DHCP authentication plug-ins are configured in the Retailer object in the directory. The SAE selects the retailer based on the domain name of the login request. If the Retailer object does not specify a DHCP authentication plug-in, the default retailer authentication plug-in is used for authentication.

If authentication fails, the SAE sends a discover decision with accept = false to the router.

3. Classifies the DHCP request.

The SAE runs a DHCP classification script to select the DHCP profile to load. If it does not find a profile, the SAE sends a discover decision with accept = false to the router.

4. Loads a DHCP profile.

The SAE loads the selected DHCP profile from the directory.

5. Authorizes the DHCP request.

The SAE authorizes the request by using the globally configured DHCP authorization plug-ins, which can include a subscriber-limiting plug-in.

Note that if the DHCP profile contains configuration parameters and the DHCP authorization plug-ins also return parameters, the plug-in parameters take precedence.

6. Assigns the address to the subscriber.

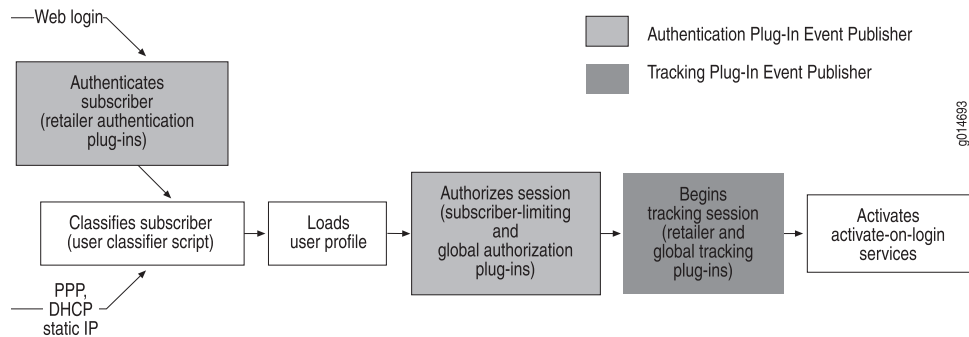
The SAE sends a DHCP discover decision to the router, which enables the router to assign an address to the subscriber. When the subscriber accepts the assigned address, the router sends an address request to the SAE, and the SAE starts processing a DHCP login request. See "Creating and Tracking Subscriber Sessions" on page 86.

- Related Topics**
- Overview of Classification Scripts on page 43
 - Activating and Tracking Service Sessions on page 87

Creating and Tracking Subscriber Sessions

Figure 19 on page 86 shows the process that the SAE uses to create and begin tracking subscriber sessions.

Figure 19: Creating and Tracking Subscriber Sessions



To create and track a subscriber session, the SAE:

1. Authenticates the login request.
 - a. Web logins are authenticated by the SAE directly. The SAE maps the login request to a retailer object in the directory by matching the requested domain name. If the retailer object:
 - Has an authentication plug-in configured, the SAE asks the plug-in to authenticate the subscriber.
 - Does not have an authentication plug-in configured, the SAE sends the authentication request to the default retailer authentication plug-in.
 - b. PPP and static IP interface addresses are authenticated by the router using the RADIUS setup configured in the router. The SAE is notified only after the authentication is completed successfully.

2. Classifies the subscriber.

The SAE runs a subscriber classification script to select the subscriber profile to load.

3. Loads a subscriber profile.

The SAE loads the selected subscriber profile from the directory.

4. Authorizes the subscriber session.

The SAE authorizes the subscriber session before it starts the session:

- a. The SAE checks the number of concurrent logins of the subscriber profile and its parent and sibling profiles and sends an event to the subscriber-limiting plug-in. If the maximum number of allowed concurrent logins configured in the plug-in is exceeded, the subscriber session is not authorized.

- b. The SAE calls the global subscriber authorization plug-in instances, which can perform custom authorization.

If any of the previous steps fail, the SAE either keeps the currently active subscriber profile (in case of a Web login) or loads the unauthenticated subscriber profile. The reason for the failure is stored in the unauthenticated profile and can be displayed when the subscriber eventually connects to the portal.

5. Sends start subscriber tracking events.

The SAE sends subscriber session start events to tracking plug-ins configured for the associated retailer and to global subscriber tracking plug-in instances.

When a subscriber session is closed, the SAE sends subscriber session stop tracking events to the same plug-ins that received the subscriber session start events.

The SAE does not create subscriber session interim update events.

6. Activates services for the subscriber that are set up to activate on login.

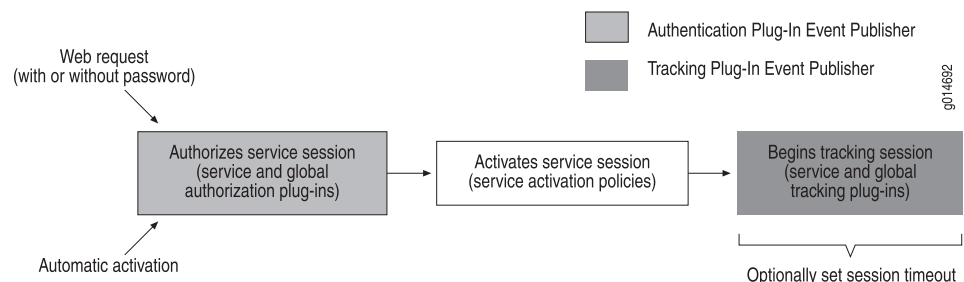
- Related Topics**
- Activating and Tracking Service Sessions on page 87
 - Tracking and Controlling Subscriber and Service Sessions with SAE APIs
 - SAE Accounting
 - Configuring Tracking Plug-Ins on page 96

Activating and Tracking Service Sessions

Figure 20 on page 87 shows the process that the SAE uses to activate and then track services. The SAE can activate services in one of two ways:

- Automatically—After the SAE creates a subscriber session, it activates all activate-on-login service subscriptions.
- Manually—Through a call of the portal application programming interface (API) method `Subscription.setActive`. This method is typically provided in the form of a Web portal and allows interaction with the subscriber.

Figure 20: Activating and Tracking Service Sessions



To activate and begin tracking a service session, the SAE:

1. Authorizes the service session.

The SAE sends events to authorization plug-in instances configured for the service and to global service authorization plug-in instances.

Service authorization plug-ins may perform authentication as well as authorization. If you define a plug-in instance to perform authentication, the portal developer must set username and password values before subscribers try to activate the service. Because the subscriber must provide the username and password, it is not possible to automatically activate a service that requires authentication.

2. Activates the services by applying service activation policies.
3. Begins tracking the service.

Sends a service session start event to the tracking plug-in instances configured for the service and to the global service tracking plug-in instances. If interim accounting is configured, a service session interim update event is sent at regular intervals to all tracking plug-ins that are registered to receive the event.

When a service is stopped (either explicitly through a call to the portal API, or implicitly through the termination of the associated subscriber session or through a timeout), a service session stop event is sent to all tracking plug-ins that received the service session start event.

Service-tracking plug-ins can set the session timeout of a service session in response to Service Session Start and Service Session Interim Update events. When a service session is active longer than the defined timeout, the SAE closes the session and sends the appropriate Service Session Stop events.

- Related Topics**
- Overview of SRC Aggregate Services
 - Creating and Tracking Subscriber Sessions on page 86
 - Configuring Tracking Plug-Ins on page 96

Chapter 6

Configuring Internal, External, and Synchronization Plug-Ins (SRC CLI)

- Configuring Internal Plug-Ins on page 89
- Configuring the SAE for External Plug-Ins on page 90
- Configuring the State Synchronization Plug-In Interface on page 91

Configuring Internal Plug-Ins

Use the following configuration statements to configure internal plug-ins:

```
shared sae configuration plug-ins name name internal {  
    plug-in-class plug-in-class ;  
}  
shared sae configuration plug-ins name name internal properties name {  
    value ;  
}
```

To configure an internal plug-in:

1. From configuration mode, access the internal plug-in configuration.

```
user@host# edit shared sae configuration plug-ins name intnl internal
```

2. Configure the Java class name of the plug-in.

```
[edit shared sae configuration plug-ins name intnl internal]  
user@host# set plug-in-class plug-in-class
```

3. Access the internal plug-in property configuration.

```
[edit shared sae configuration plug-ins name intnl internal]  
user@host# edit properties prop
```

4. Configure properties that define the plug-in. Enter values in the format property name = expression.

```
[edit shared sae configuration plug-ins name internalPlugin internal properties  
prop]  
user@host# set value
```

- Related Topics**
- Configuring Internal Plug-Ins (C-Web Interface)
 - Configuring the SAE for External Plug-Ins on page 90
 - How Internal Plug-Ins Work on page 81
 - Types of Internal Plug-Ins on page 82

Configuring the SAE for External Plug-Ins

You need to configure SAE external plug-ins for SAE plug-in agents in the NIC, for Admission Control Plug-Ins, and for custom plug-ins developed in Common Object Request Broker Architecture (CORBA). For information about external plug-ins, see SAE Plug-Ins .

When you use an external plug-in, you need to export its object reference to the SAE. When the SAE sends the first event to a registered plug-in, it resolves the object reference. In case of a failure, the SAE resolves the object reference again. In this case, if a plug-in restarts and instantiates a different object (that is, a different object reference), the SAE learns about the new object through the naming service or the file reference.

You can configure the SAE to resolve the object reference and specify which attributes to send to the external plug-in. To do so with the SRC CLI, use the following configuration statements:

```
shared sae configuration plug-ins name name external {
  corba-object-reference corba-object-reference ;
  attr [(host | router-name | interface-name | interface-alias | interface-descr | port-id
    | user-ip-address | login-name | accounting-id | auth-user-id | if-radius-class |
    if-session-id | service-name | radius-class | event-time | session-id | terminate-cause
    | session-time | in-octets | out-octets | in-packets | out-packets | nas-ip |
    user-mac-address | service-session-name | service-session-tag | user-type |
    user-radius-class | user-session-id | primary-user-name | subscription-name | login-id
    | if-index | event-time-millisecond | nas-port | operational | user-inet-address |
    nas-inet-address | router-type | interface-speed | service-bundle | user-dn | uid |
    domain | retailer-dn | password | service-scope | session-timeout |
    downstream-bandwidth | upstream-bandwidth | dhcp-packet | aggr-session-id |
    aggr-login-name | aggr-user-dn | aggr-user-inet-address | aggr-accounting-id |
    aggr-auth-user-id)...];
}
```

To configure an external plug-in:

1. From configuration mode, access the external plug-in configuration.


```
user@host# edit shared sae configuration plug-ins name NicAgent external
```
2. Configure the object reference of the external plug-in that is exported to the SAE.


```
[edit shared sae configuration plug-ins name NicAgent external]
user@host# set corba-object-reference corba-object-reference
```
3. Configure the attributes that are sent to the external plug-in.

```
[edit shared sae configuration plug-ins name NicAgent external]
user@host# set attr [(host | router-name | interface-name | interface-alias | ...)...]
```

4. (Optional) Verify your configuration.

```
[edit shared sae configuration plug-ins name NicAgent external]
user@host# show

corba-object-reference corbaloc:boston:8801/nic;
attributes [ router-name router-type interface-descr interface-speed
service-bundle ];
```

- Related Topics**
- Configuring Internal Plug-Ins on page 89
 - Configuring the SAE for External Plug-Ins (C-Web Interface)
 - Configuring the State Synchronization Plug-In Interface on page 91
 -

Configuring the State Synchronization Plug-In Interface

Some external plug-ins, such as the Admission Control Plug-In (ACP) application and the SAE plug-in agent for the NIC, support state synchronization with the SAE. The state synchronization plug-in interface allows external plug-ins to maintain the state of active subscriber, service, and interface sessions without having to store intermediate versions of the state locally.

Use the following configuration statements to configure the state synchronization plug-in:

```
shared sae configuration plug-ins state-synchronization {
  fail-queue-size fail-queue-size ;
  fail-queue-age fail-queue-age ;
  batch-time batch-time ;
  keepalive-time keepalive-time ;
}
shared sae configuration plug-ins manager {
  threads threads ;
}
```

To configure the state synchronization plug-in interface:

1. From configuration mode, access the state synchronization plug-in configuration.

```
user@host# edit shared sae configuration plug-ins state-synchronization
```

2. Configure the maximum number of plug-in events that are stored while the communication with a state synchronization plug-in is interrupted.

```
[edit shared sae configuration plug-ins state-synchronization]
user@host# set fail-queue-size fail-queue-size
```

3. Configure the maximum time that plug-in events are stored while the communication with a state synchronization plug-in is interrupted.

```
[edit shared sae configuration plug-ins state-synchronization]
user@host# set fail-queue-age fail-queue-age
```

4. Configure the time that the SAE waits for other plug-ins to become ready before starting a synchronization sequence.

```
[edit shared sae configuration plug-ins state-synchronization]
user@host# set batch-time batch-time
```

5. Configure the time that the SAE waits after an event before sending a ping to the remote plug-in.

```
[edit shared sae configuration plug-ins state-synchronization]
user@host# set keepalive-time keepalive-time
```

6. Configure the number of threads that the SAE maintains for plug-in synchronization.

```
[edit shared sae configuration plug-ins state-synchronization]
user@host# up
user@host# [edit shared sae configuration plug-ins]
user@host# set manager threads 5
```

7. (Optional) Verify your configuration.

```
[edit shared sae configuration plug-ins state-synchronization]
user@host# show
fail-queue-size 5000;
fail-queue-age -1;
batch-time 60;
keepalive-time 60;

user@host# [edit shared sae configuration plug-ins]
user@host# show
threads 5;
```

- Related Topics**
- Configuring the State Synchronization Plug-In Interface (C-Web Interface)
 - Configuring the SAE for External Plug-Ins on page 90
 - SAE Plug-Ins
 -

Chapter 7

Configuring Accounting and Authentication Plug-Ins (SRC CLI)

- Creating RADIUS Peers on page 93
- Types of Tracking Plug-Ins on page 95
- Configuring Tracking Plug-Ins on page 96
- Types of Authentication Plug-Ins on page 106
- Configuring Authentication Plug-Ins on page 107
- Configuring UDP Ports for RADIUS Plug-Ins on page 117
- Defining RADIUS Packets for Flexible RADIUS Plug-Ins on page 118
- Configuring Event Publishers on page 130

Creating RADIUS Peers

RADIUS peers are instances of RADIUS servers. If you define multiple servers, the SAE uses them in cases of failover or as alternate routers for load-balancing purposes.

Each RADIUS plug-in requires a default peer. Configure a RADIUS peer before you configure the plug-in.

RADIUS peers are configured in the peer group for each RADIUS plug-in. Use the following configuration statements to configure a RADIUS peer:

```
shared sae configuration plug-ins name name radius-accounting peer-group name
{
    server-address server-address ;
    server-port server-port ;
    secret secret ;
}
shared sae configuration plug-ins name name radius-authentication peer-group name
{
    server-address server-address ;
    server-port server-port ;
    secret secret ;
}
shared sae configuration plug-ins name name custom-radius-accounting peer-group
name {
    server-address server-address ;
    server-port server-port ;
}
```

```

    secret secret ;
}
shared sae configuration plug-ins name name custom-radius-authentication peer-group
name {
    server-address server-address ;
    server-port server-port ;
    secret secret ;
}
shared sae configuration plug-ins name name flex-radius-accounting peer-group
name {
    server-address server-address ;
    server-port server-port ;
    secret secret ;
}
shared sae configuration plug-ins name name flex-radius-authentication peer-group
name {
    server-address server-address ;
    server-port server-port ;
    secret secret ;
}

```

To create a RADIUS peer:

1. From configuration mode, access the RADIUS peer configuration for the plug-in that you are configuring. In this sample procedure, the RADIUS peer is configured in the west-region SAE group.

```

user@host# edit shared sae group west-region configuration plug-ins name
basicRadius radius-accounting peer-group peer1

```

2. Configure the IP address of the RADIUS server to which the SAE sends accounting data.

```

[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting peer-group peer1]
user@host# set server-address server-address

```

3. Configure the port used for RADIUS packets.

```

[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting peer-group peer1]
user@host# set server-port server-port

```

4. Configure the password that is shared with the RADIUS server. You must configure the same password on the RADIUS server.

```

[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting peer-group peer1]
user@host# set secret secret

```

5. (Optional) Verify your configuration.

```

[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting peer-group peer1]

```

```

user@host# show
server-address 10.10.1.1;
server-port 1812;
secret *****;

```

- Related Topics**
- Creating Grouped Configurations for the SAE (SRC CLI)
 - Using Flexible RADIUS Packet Definitions on page 127
 - Defining the Values of RADIUS Attributes on page 121
 - Configuring a RADIUS Packet Template on page 125

Types of Tracking Plug-Ins

You can configure the tracking plug-ins described in Table 9 on page 95.

By default, the fileAcct plug-in instance tracks all subscriber and service sessions and writes all available attributes to a file. You can use this plug-in instance or create new one.



NOTE: When you use the NAS-Port attribute in tracking plug-ins, the SAE calculates the NAS-Port value based on the NAS-Port-Id value that it receives from the JUNOSe router. You can change the NAS-Port format in the JUNOSe software. However, because the SAE has no indication of which format is configured on the JUNOSe router, the calculation of the NAS-Port attribute is correct only if the router uses the default configuration.

Table 9: Tracking Plug-Ins

Plug-In	Description
Basic RADIUS accounting	<p>Sends accounting information to an external RADIUS accounting server or a group of redundant servers.</p> <p>Java class name—net.juniper.smgt.sae.plugin.RadiusTrackingPluginEventListener</p>
Custom RADIUS accounting	<p>Provides customized functions that can also be found in the flexible RADIUS accounting plug-ins. Custom plug-ins are internal plug-ins that are designed to deliver better system performance than the flexible RADIUS plug-ins. You can extend this plug-in by using the RADIUS client library.</p> <p>Java class name—net.juniper.smgt.sae.plugin.CustomRadiusAccounting</p>
Flat file accounting	<p>Writes tracking information to a file in comma-separated format.</p> <p>Java class name—net.juniper.smgt.sae.plugin.FileTrackingPluginEventListener</p>

Table 9: Tracking Plug-Ins *(continued)*

Plug-In	Description
Flexible RADIUS accounting	<p>Performs the same functions as the basic RADIUS accounting plug-in, but also lets you customize RADIUS accounting packets that the SAE sends to RADIUS servers. You can specify which fields are included in RADIUS accounting packets and what information is contained in the fields.</p> <p>Java class name—<code>net.juniper.smgmt.sae.plugin.FlexibleRadiusTrackingPluginEventListener</code></p>
PCMM record-keeping server plug-in	<p>Sends accounting information to an external PCMM record-keeping server (RKS). See Configuring PCMM Record-Keeping Server Plug-Ins with SRC CLI.</p> <p>Java class name—<code>net.juniper.smgmt.sae.plugin.RksEventListener</code></p>
QoS profile tracking	<p>Ensures that as a subscriber activates and deactivates services, the correct QoS profile is attached to the subscriber interface. See Dynamically Managing QoS Profiles.</p> <p>Java class name—<code>net.juniper.smgmt.sae.plugin.qtp.QosProfileTrackingPluginEventListener</code></p>

Related Topics

- [Types of Internal Plug-Ins on page 82](#)
- [Creating and Tracking Subscriber Sessions on page 86](#)
- [Activating and Tracking Service Sessions on page 87](#)
- [Configuring Tracking Plug-Ins on page 96](#)

Configuring Tracking Plug-Ins

You can perform the following tasks to configure tracking plug-ins:

- [Configuring Flat File Accounting Plug-Ins on page 96](#)
- [Configuring Headers for Flat File Accounting Plug-Ins on page 98](#)
- [Configuring Basic RADIUS Accounting Plug-Ins on page 99](#)
- [Configuring Flexible RADIUS Accounting Plug-Ins on page 101](#)
- [Configuring Custom RADIUS Accounting-Plug-Ins on page 104](#)

Configuring Flat File Accounting Plug-Ins

Flat file accounting plug-ins write information to a file in a comma-separated format. The SRC software has a default flat file accounting plug-in instance called `fileAcct`. The `fileAcct` instance logs all possible attributes for 24-hour periods in the file `var/acct/log`.

Another item that you can configure for flat files is the names of the headers that appear in the file.

Use the following configuration statements to create flat-file accounting plug-in instances:

```
shared sae configuration plug-ins name name file-accounting {
  filename filename ;
  template template ;
  interval interval ;
  fields [(status | nas-id | host | router-name | interface-name | interface-alias |
    interface-descr | port-id | user-ip-address | login-name | accounting-id | auth-user-id
    | if-radius-class | if-session-id | service-name | radius-class | event-time | session-id
    | terminate-cause | session-time | in-octets | out-octets | in-packets | out-packets
    | nas-ip | user-mac-address | service-session-name | service-session-tag | user-type
    | user-radius-class | user-session-id | primary-user-name | subscription-name |
    login-id | if-index | event-time-millisecond | nas-port | operational | user-inet-address
    | nas-inet-address | router-type | interface-speed)...];
}
```

To create flat-file accounting plug-ins:

1. From configuration mode, access the basic RADIUS accounting plug-in configuration. In this sample procedure, the plug-in called fileAcct is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name
fileAcct file-accounting
```

2. Configure the name and location of the file to which the SAE writes accounting information.

```
[edit shared sae group west-region configuration plug-ins name fileAcct
file-accounting]
user@host# set filename filename
```

3. Configure the name of the template that defines header names for attributes listed in accounting files.

```
[edit shared sae group west-region configuration plug-ins name fileAcct
file-accounting]
user@host# set template template
```

4. Configure the number of hours of information stored in each accounting file.

```
[edit shared sae group west-region configuration plug-ins name fileAcct
file-accounting]
user@host# set interval interval
```

5. Configure the fields that you want to record in the accounting file.

```
[edit shared sae group west-region configuration plug-ins name fileAcct
file-accounting]
user@host# set fields [(status | nas-id | host | router-name | interface-name |
interface-alias | interface-descr | port-id | user-ip-address | login-name |
accounting-id | auth-user-id | if-radius-class | if-session-id | service-name |
radius-class | event-time | session-id | terminate-cause | session-time | in-octets
| out-octets | in-packets | out-packets | nas-ip | user-mac-address |
```

```

service-session-name | service-session-tag | user-type | user-radius-class |
user-session-id | primary-user-name | subscription-name | login-id | if-index |
event-time-millisecond | nas-port | operational | user-inet-address |
nas-inet-address | router-type | interface-speed)...]

```

6. (Optional) Verify your configuration.

```

[edit shared sae group west-region configuration plug-ins name fileAcct
file-accounting]
user@host# show
filename var/acct/log;
template FileAccounting.std;
interval 24;
fields [ status nas-id host router-name interface-name interface-alias
interface-descr port-id user-inet-address login-name accounting-id
auth-user-id if-session-id service-name event-time session-id
terminate-cause session-time in-octets out-octets in-packets out-packets
nas-inet-address user-mac-address service-session-name service-session-tag
user-type user-session-id ];

```

Configuring Headers for Flat File Accounting Plug-Ins

When the SAE writes data to a flat file, it writes into the first line the headers that identify the attributes in the file. For example, in the following accounting file, the first line lists headers for all attribute fields in the file, and the following lines list the actual data in each field:

```

Accounting Status,NAS ID,SSP Host,Router Name,Interface Name,Interface
Alias,Interface Description,NAS port ID,User IP Address,User ID,User Accounting
ID,User Authentication ID,INTF Radius Class,INTF,SessionId, Service Name,Radius
Class,Timestamp,SessionId, Terminate Cause,Session Time,Input Octets,Output
Octets,Input Packets,Output Packets,NAS IP,User Mac address,Service Session
Name,Service Session Tag,User Session Type,User Session Radius Class,User
Session ID
start,SSP.uelmo,uelmo,default@erx7_ssp57,FastEthernet1/1.1,,IP1/1.1,default@erx7_ssp57
FastEthernet1/1:65535, 10.10.10.20,pebbles@virneo.net,,,erx fastEthernet
1/1:0001048619,Video-Gold,Video-Gold,Fri Jan 30 14:23:29 EDT 2004,
VideoGold:null:1064946209182, 0,0,0,0,0, 10.10.7.17,,,PPP,,
pebbles:1064946144841

```

You can assign your own names to the headers that appear in the file. To do so, define the header names in a template, and then set up file accounting plug-in instances to use the template. The default template, `FileAccounting.std`, defines header names for all possible attributes. You can use the default template or create your own templates.

Use the following configuration statements to create a file accounting template:

```

shared sae configuration file-accounting-template name ...
shared sae configuration file-accounting-template name attributes (status | nas-id |
host | router-name | interface-name | interface-alias | interface-descr | port-id |
user-ip-address | login-name | accounting-id | auth-user-id | if-radius-class | if-session-id
| service-name | radius-class | event-time | session-id | terminate-cause | session-time
| in-octets | out-octets | in-packets | out-packets | nas-ip | user-mac-address |

```

```

service-session-name | service-session-tag | user-type | user-radius-class |
user-session-id | primary-user-name | subscription-name | login-id | if-index |
event-time-millisecond | nas-port | operational | user-inet-address | nas-inet-address
| router-type | interface-speed | service-bundle | user-dn | uid | domain | retailer-dn |
password | service-scope | session-timeout | downstream-bandwidth |
upstream-bandwidth | dhcp-packet | aggr-session-id | aggr-login-name | aggr-user-dn
| aggr-user-inet-address | aggr-accounting-id | aggr-auth-user-id) {
value ;
}

```

To set up a file accounting template:

1. From configuration mode, access the file accounting template configuration. In this sample procedure, the template called `std` is configured in the `west-region` SAE group.

```

user@host# edit shared sae group west-region configuration
file-accounting-template std

```

2. Define header names.

```

[edit shared sae group west-region configuration file-accounting-template std]
user@host# set attributes attribute value

```

For example:

```

[edit shared sae group west-region configuration file-accounting-template std]
user@host# set attributes terminate-cause "RADIUS Termination Cause"

```

3. (Optional) Verify your configuration.

```

[edit shared sae group west-region configuration file-accounting-template
std]
user@host# show
attributes {
    terminate-cause "RADIUS Termination Cause";
    service-session-name "Service Session Name";
}

```

Configuring Basic RADIUS Accounting Plug-Ins

You can use basic RADIUS accounting plug-ins to send accounting information to an external RADIUS accounting server or to a group of redundant servers. To communicate with nonredundant servers, you need to create multiple instances of the plug-in.

Use the following configuration statements to configure RADIUS accounting plug-ins:

```

shared sae configuration plug-ins name name radius-accounting {
    load-balancing-mode (failover | roundRobin);
    fallback-timer fallback-timer ;
    nas-ip (Ssplp | Erxlp);
    retry-interval retry-interval ;
}

```

```

maximum-queue-length maximum-queue-length ;
bind-address bind-address ;
udp-port udp-port ;
username (login-name | accounting-id | auth-user-name | manager-id);
calling-station-id (mac | no);
default-peer default-peer ;
}

```

To set up basic RADIUS accounting plug-ins:

1. From configuration mode, access the basic RADIUS accounting plug-in configuration. In this sample procedure, the plug-in called basicRadius is configured in the west-region SAE group.

```

user@host# edit shared sae group west-region configuration plug-ins name
basicRadius radius-accounting

```

2. Configure the mode for load-balancing RADIUS servers.

```

[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
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 basicRadius
radius-accounting]
user@host# set fallback-timer fallback-timer

```

4. (Optional) Configure the value of the NAS-IP attribute.

```

[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set nas-ip (Ssplp | Erxlp)

```

5. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```

[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set retry-interval retry-interval

```

6. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```

[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set maximum-queue-length maximum-queue-length

```

7. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```

[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set bind-address bind-address

```

8. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set udp-port udp-port
```

9. Configure the value of the User-Name attribute (RADIUS attribute [1]).

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set username (login-name | accounting-id | auth-user-name |
manager-id)
```

10. Specify whether the SAE sends the MAC address of the subscriber in the Calling-Station-Id attribute.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set calling-station-id (mac | no)
```

11. Configure the default peer, which is the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set default-peer default-peer
```

12. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# show
load-balancing-mode failover;
failback-timer -1;
retry-interval 3000;
maximum-queue-length 10000;
username login-name;
calling-station-id no;
default-peer peer1;
```

Configuring Flexible RADIUS Accounting Plug-Ins

Flexible RADIUS accounting plug-ins provide the same features as basic RADIUS accounting plug-ins. In addition, they allow you to customize RADIUS accounting packets that the SAE sends to RADIUS servers. You can specify which fields are included in the RADIUS accounting packets and what information is contained in the fields.

Use the following configuration statements to configure flexible RADIUS accounting plug-ins:

```
shared sae configuration plug-ins name name flex-radius-accounting {
```

```

load-balancing-mode (failover | roundRobin);
failback-timer failback-timer ;
timeout timeout ;
retry-interval retry-interval ;
maximum-queue-length maximum-queue-length ;
bind-address bind-address ;
udp-port udp-port ;
error-handling (0 | 1);
default-peer default-peer ;
template template ;
}

```

To set up flexible RADIUS accounting plug-ins:

1. From configuration mode, access the flexible RADIUS accounting plug-in configuration. In this sample procedure, the plug-in called flexRadiusAct is configured in the west-region SAE group.

```

user@host# edit shared sae group west-region configuration plug-ins name
flexRadiusAct flex-radius-accounting

```

2. Configure the mode for load-balancing RADIUS servers.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
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 flexRadiusAct
flex-radius-accounting]
user@host# set failback-timer failback-timer

```

4. (Optional) Configure the maximum time the SAE waits for a response from a RADIUS server.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
user@host# set timeout timeout

```

5. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
user@host# set retry-interval retry-interval

```

6. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
user@host# set maximum-queue-length maximum-queue-length

```

7. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
user@host# set bind-address bind-address
```

8. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
user@host# set udp-port udp-port
```

9. Configure the way the SAE handles errors.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
user@host# set error-handling (0 | 1)
```

10. Configure the name of the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
user@host# set default-peer default-peer
```

11. Configure the name of the RADIUS packet template that defines attributes for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
user@host# set template template
```

12. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name
flexRadiusAct flex-radius-accounting]
user@host# show
load-balancing-mode failover;
failback-timer -1;
timeout 15000;
retry-interval 3000;
maximum-queue-length 10000;
error-handling 0;
default-peer peer2;
template stdAcct;
peer-group peer2 {
  server-address 10.10.1.1;
  server-port 1818;
  secret *****;
}
```

Configuring Custom RADIUS Accounting-Plug-Ins

The custom RADIUS accounting plug-ins provide the same functions as the flexible RADIUS accounting plug-ins, but are designed to deliver better system performance. To use a custom plug-in, you must provide a Java class that implements the service provider interface (SPI) defined in the RADIUS client library. Use this SPI to specify which fields and field values to include in RADIUS accounting packets. The RADIUS client library is part of the SAE core application programming interface (API).

See the documentation for the RADIUS client library 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 `SDK+AppSupport+Demos+Samples.tar.gz` file on the Juniper Networks Web site at: <https://www.juniper.net/support/csc/swdist-erx/src.html>. The application is located the following directory:

SDK/plugin/java/src/net/juniper/smg/sample/radiuslib/RadiusPacketHandlerImpl.java.

Use the following configuration statements to set up custom RADIUS accounting plug-ins:

```
shared sae configuration plug-ins name name custom-radius-accounting {
  java-class-radius-packet-handler java-class-radius-packet-handler ;
  class-path-radius-packet-handler class-path-radius-packet-handler ;
  append-acct-status-type-attribute;
  require-mandatory-attributes;
  load-balancing-mode (failover | roundRobin);
  fallback-timer fallback-timer ;
  timeout timeout ;
  retry-interval retry-interval ;
  maximum-queue-length maximum-queue-length ;
  bind-address bind-address ;
  udp-port udp-port ;
  default-peer default-peer ;
}
```

To set up custom RADIUS accounting plug-ins:

1. From configuration mode, access the custom RADIUS accounting plug-in configuration. In this sample procedure, the plug-in called `customRadiusAct` is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name  
customRadiusAct custom-radius-accounting
```

2. Configure the name of the Java class that implements the `RadiusPacketHandler` interface in the RADIUS client library.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct  
custom-radius-accounting]  
user@host# set java-class-radius-packet-handler java-class-radius-packet-handler
```


3. Configure the URLs that identify a location from which Java classes are loaded when the plug-in is initialized.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set class-path-radius-packet-handler class-path-radius-packet-handler
```

4. (Optional) Enable the plug-in to include the Acct-Status-Type attribute in a RADIUS accounting request packet.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set append-acct-status-type-attribute
```

5. (Optional) Specify that a RADIUS authentication or accounting request must contain all mandatory RADIUS attributes before sending the request packet.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set require-mandatory-attributes
```

6. Configure the mode for load-balancing RADIUS servers.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set load-balancing-mode (failover | roundRobin)
```

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

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set failback-timer failback-timer
```

8. (Optional) Configure the maximum time the SAE waits for a response from a RADIUS server.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set timeout timeout
```

9. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set retry-interval retry-interval
```

10. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set maximum-queue-length maximum-queue-length
```

11. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set bind-address bind-address
```

12. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set udp-port udp-port
```

13. Configure the name of the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set default-peer default-peer
```

14. (Optional) From operational mode, verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name
 customRadiusAct custom-radius-accounting]
user@host# show
java-class-radius-packet-handler
net.juniper.smgmt.radius.RadiusPacketHandlerImpl;
append-acct-status-type-attribute;
load-balancing-mode failover;
failback-timer -1;
timeout 15000;
retry-interval 3000;
maximum-queue-length 10000;
default-peer peer3;
```

- Related Topics**
- Activating and Tracking Service Sessions on page 87
 - Types of Tracking Plug-Ins on page 95

Types of Authentication Plug-Ins

This section shows how to configure the authentication plug-ins described in Table 10 on page 107. Because authentication and authorization are similar, the plug-in user interface does not distinguish between them. However, when you configure plug-ins, you need to set them up to perform the correct behavior, either authentication or authorization.

You can configure multiple authentication plug-ins. The plug-ins are called in an arbitrary order, and each plug-in can return authorization values. (If multiple plug-ins

return a session-timeout value, the smallest value is used.) Authentication or authorization succeeds if all plug-in calls succeed.

Table 10: Authentication Plug-Ins

Plug-In	Description
Basic RADIUS authentication	<p>Sends authentication information to an external RADIUS authentication server or a group of redundant servers.</p> <p>Java class name—<code>net.juniper.smgmt.sae.plugin.RadiusAuthPluginEventListener</code></p>
Custom RADIUS authentication	<p>Provides customized functions that can also be found in the flexible RADIUS authentication plug-ins. Custom plug-ins are internal plug-ins that are designed to deliver better system performance than the flexible RADIUS plug-ins. You can extend this plug-in by using the RADIUS client library.</p> <p>Java class name—<code>net.juniper.smgmt.sae.plugin.CustomRadiusAuth</code></p>
Flexible RADIUS authentication	<p>Performs the same functions as the basic RADIUS authentication plug-in, but also lets you customize RADIUS authentication packets that the SAE sends to RADIUS servers. You can specify which fields are included in RADIUS authentication packets and what information is contained in the fields.</p> <p>Java class name—<code>net.juniper.smgmt.sae.plugin.FlexibleRadiusAuthPluginEventListener</code></p>
LDAP authentication	<p>Performs authentication against different directories using different authentication methods. There are two LDAP authentication plug-ins: one authenticates subscribers, and the second authenticates SRC administrators so that they can access the SAE Web Admin application.</p> <p>Java class name of the subscriber authentication plug-in—<code>net.juniper.smgmt.sae.plugin.LdapAuthenticator</code></p> <p>Java class name of the administrator authentication plug-in—<code>net.juniper.smgmt.sae.plugin.adminLdap</code></p>
Limiting subscribers	<p>Limits the number of authenticated subscribers who connect to an IP interface on the router.</p> <p>Java class name—<code>net.juniper.smgmt.sae.plugin.LimitNumSubscriberPerIntfAuthPluginListener</code></p>

- Related Topics**
- Types of Internal Plug-Ins on page 82
 - How Internal Plug-Ins Work on page 81
 - Configuring Authentication Plug-Ins on page 107

Configuring Authentication Plug-Ins

You can perform the following tasks to configure authentication plug-ins:

1. Limiting Subscribers on Router Interfaces on page 108
2. Configuring Basic RADIUS Authentication Plug-Ins on page 108
3. Configuring Flexible RADIUS Authentication Plug-Ins on page 110

4. Configuring Custom RADIUS Authentication Plug-Ins on page 112
5. Configuring LDAP Authentication Plug-Ins on page 115

Limiting Subscribers on Router Interfaces

You can limit the number of authenticated subscribers who connect to an IP interface on the router. This plug-in does not limit the number of unauthenticated subscribers who connect to an IP interface, and does not limit the number of subscribers who connect to a physical or link-layer interface. In the case of subscriber interfaces, the plug-in limits the number of authenticated subscribers on the subscriber interface but not on the underlying primary IP interface.

Use the following configuration statement to set up a plug-in that limits the number of subscribers who connect to interfaces:

```
shared sae configuration plug-ins name name interface-subscriber-limit {
    concurrent-subscribers concurrent-subscribers ;
}
```

To set up a plug-in that limits the number of subscribers on interfaces:

1. From configuration mode, access the custom RADIUS accounting plug-in configuration. In this sample procedure, the plug-in called subsLimit is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name
subsLimit interface-subscriber-limit
```

2. Configure the number of authenticated subscribers who can connect to an IP interface on the router simultaneously.

```
[edit shared sae group west-region configuration plug-ins name subsLimit
 interface-subscriber-limit]
user@host# set concurrent-subscribers concurrent-subscribers
```

3. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name subsLimit
 interface-subscriber-limit]
user@host# show
concurrent-subscribers 1;
```

Configuring Basic RADIUS Authentication Plug-Ins

You can use basic RADIUS authentication plug-ins to send authentication information to an external RADIUS accounting server or a group of redundant servers. To communicate with nonredundant servers, you need to create additional instances of the plug-in.

Use the following configuration statements to set up basic RADIUS authentication plug-ins:

```

shared sae configuration plug-ins name name radius-authentication {
  load-balancing-mode (failover | roundRobin);
  fallback-timer fallback-timer ;
  nas-ip (Ssplp | Erxlp);
  retry-interval retry-interval ;
  maximum-queue-length maximum-queue-length ;
  bind-address bind-address ;
  udp-port udp-port ;
  default-peer default-peer ;
}

```

To set up basic RADIUS authentication plug-ins:

1. From configuration mode, access the basic RADIUS authentication plug-in configuration. In this sample procedure, the plug-in called RadiusAuth is configured in the west-region SAE group.

```

user@host# edit shared sae group west-region configuration plug-ins name
RadiusAuth radius-authentication

```

2. Configure the mode for load-balancing RADIUS servers.

```

[edit shared sae group west-region configuration plug-ins name RadiusAuth
 radius-authentication]
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 RadiusAuth
 radius-authentication]
user@host# set fallback-timer fallback-timer

```

4. (Optional) Configure the value of the NAS-Ip attribute.

```

[edit shared sae group west-region configuration plug-ins name RadiusAuth
 radius-authentication]
user@host# set nas-ip (Ssplp | Erxlp)

```

5. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```

[edit shared sae group west-region configuration plug-ins name RadiusAuth
 radius-authentication]
user@host# set retry-interval retry-interval

```

6. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```

[edit shared sae group west-region configuration plug-ins name RadiusAuth
 radius-authentication]
user@host# set maximum-queue-length maximum-queue-length

```

7. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth
 radius-authentication]
user@host# set bind-address bind-address
```

8. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth
 radius-authentication]
user@host# set udp-port udp-port
```

9. Configure the name of the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth
 radius-authentication]
user@host# set default-peer default-peer
```

10. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name RadiusAuth
 radius-authentication]
user@host# show
load-balancing-mode failover;
failback-timer -1;
retry-interval 3000;
maximum-queue-length 10000;
default-peer peer1;
```

Configuring Flexible RADIUS Authentication Plug-Ins

Flexible RADIUS authentication plug-ins provide the same features as basic RADIUS authentication plug-ins. In addition, they allow you to customize RADIUS authentication packets that the system sends to RADIUS servers and specify which fields are included in the RADIUS authentication packets and what information is contained in the fields.

Use the following configuration statements to set up flexible RADIUS authentication plug-ins:

```
shared sae configuration plug-ins name name flex-radius-authentication {
  load-balancing-mode (failover | roundRobin);
  failback-timer failback-timer ;
  timeout timeout ;
  retry-interval retry-interval ;
  maximum-queue-length maximum-queue-length ;
  bind-address bind-address ;
  udp-port udp-port ;
```

```

error-handling (0 | 1);
default-peer default-peer;
template template ;
}

```

To set up flexible RADIUS authentication plug-ins:

1. From configuration mode, access the flexible RADIUS authentication plug-in configuration. In this sample procedure, the plug-in called flexRadiusAuth is configured in the west-region SAE group.

```

user@host# edit shared sae group west-region configuration plug-ins name
flexRadiusAuth flex-radius-authentication

```

2. Configure the mode for load-balancing RADIUS servers.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
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 flexRadiusAuth
flex-radius-authentication]
user@host# set fallback-timer fallback-timer

```

4. (Optional) Configure the maximum time the SAE waits for a response from a RADIUS server.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set timeout timeout

```

5. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set retry-interval retry-interval

```

6. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set maximum-queue-length maximum-queue-length

```

7. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```

[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set bind-address bind-address

```

8. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set udp-port udp-port
```

9. Configure the way the SAE handles errors.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set error-handling (0 | 1)
```

10. Configure the name of the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAuth
flex-radius-authentication]
user@host# set default-peer default-peer
```

11. Configure the name of the RADIUS packet template that defines attributes for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
user@host# set template template
```

12. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name
flexRadiusAuth flex-radius-authentication]
user@host# show
load-balancing-mode failover;
failback-timer -1;
timeout 15000;
retry-interval 3000;
maximum-queue-length 10000;
error-handling 0;
default-peer 1;
template stdAuth;
peer-group 1 {
    server-address ;
    server-port 1812;
    secret *****;
}
```

Configuring Custom RADIUS Authentication Plug-Ins

The custom RADIUS authentication plug-ins provide the same functions as the flexible RADIUS authentication plug-ins, but are designed to deliver better system performance. To use a custom plug-in, you must provide a Java class that implements the SPI defined in the RADIUS client library. Use this SPI to specify which fields and

field values to include in RADIUS accounting packets. The RADIUS client library is part of the SAE core API.

See the documentation for the RADIUS client library 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 in the `SDK+AppSupport+Demos+Samples.tar.gz` file on the Juniper Networks Web site at:

<https://www.juniper.net/support/csc/swdist-erx/src.html> The application is located the following directory:

`SDK/plugin/java/src/net/juniper/smgmt/sample/radiuslib/RadiusPacketHandlerImpl.java`.

Use the following configuration statements to set up custom RADIUS authentication plug-ins:

```
shared sae configuration plug-ins name name custom-radius-authentication {
  java-class-radius-packet-handler java-class-radius-packet-handler ;
  class-path-radius-packet-handler class-path-radius-packet-handler ;
  require-mandatory-attributes;
  load-balancing-mode (failover | roundRobin);
  failback-timer failback-timer ;
  timeout timeout ;
  retry-interval retry-interval ;
  maximum-queue-length maximum-queue-length ;
  bind-address bind-address ;
  udp-port udp-port ;
  default-peer default-peer;
}
```

To set up custom RADIUS authentication plug-ins:

1. From configuration mode, access the custom RADIUS authentication plug-in configuration. In this sample procedure, the plug-in called `customRadiusAuth` is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name  
customRadiusAuth custom-radius-authentication
```

2. Configure the name of the Java class that implements the `RadiusPacketHandler` interface in the RADIUS client library.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth  
custom-radius-authentication]  
user@host# set java-class-radius-packet-handler java-class-radius-packet-handler
```

3. Configure the URLs that identify a location from which Java classes are loaded when the plug-in is initialized.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth  
custom-radius-authentication]  
user@host# set class-path-radius-packet-handler class-path-radius-packet-handler
```

4. (Optional) Specify that a RADIUS authentication or accounting request must contain all mandatory RADIUS attributes before sending the request packet.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
 custom-radius-authentication]
user@host# set require-mandatory-attributes
```

5. Configure the mode for load-balancing RADIUS servers.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
 custom-radius-authentication]
user@host# set load-balancing-mode (failover | roundRobin)
```

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

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
 custom-radius-authentication]
user@host# set fallback-timer fallback-timer
```

7. (Optional) Configure the maximum time the SAE waits for a response from a RADIUS server.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
 custom-radius-authentication]
user@host# set timeout timeout
```

8. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
 custom-radius-authentication]
user@host# set retry-interval retry-interval
```

9. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
 custom-radius-authentication]
user@host# set maximum-queue-length maximum-queue-length
```

10. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
 custom-radius-authentication]
user@host# set bind-address bind-address
```

11. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
custom-radius-authentication]
user@host# set udp-port udp-port
```

12. Configure the name of the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAuth
custom-radius-authentication]
user@host# set default-peer default-peer
```

13. (Optional) From operational mode, verify your configuration.

```
[edit shared sae configuration plug-ins name customRadiusAuth
custom-radius-authorization]
user@host# show
java-class-radius-packet-handler
net.juniper.smgmt.radius.RadiusPacketHandlerImpl;
require-mandatory-attributes;
load-balancing-mode failover;
failback-timer -1;
timeout 15000;
retry-interval 3000;
maximum-queue-length 10000;
default-peer peer4;
```

Configuring LDAP Authentication Plug-Ins

Use the following configuration statements to configure LDAP authentication plug-ins:

```
shared sae configuration plug-ins name name ldap-authentication {
  method (search | bind);
  server server ;
  bind-dn bind-dn ;
  bind-password bind-password ;
  search-filter search-filter ;
  (ldaps);
  search-base-dn search-base-dn ;
  name-attribute name-attribute ;
  password-attribute password-attribute ;
  service-bundle-attribute service-bundle-attribute ;
  session-volume-quota session-volume-quota ;
  timeout timeout ;
}
```

To create LDAP authentication plug-ins:

1. From configuration mode, access the custom LDAP authentication plug-in configuration. In this sample procedure, the plug-in called `ldapAuth` is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name
ldapAuth ldap-authentication
```

2. Configure the LDAP authentication method that the SAE uses.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
  ldap-authentication]
user@host# set method (search | bind)
```

3. (Optional) Configure a comma-separated list of IP addresses or hostnames of the LDAP authentication server.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
  ldap-authentication]
user@host# set server server
```

4. (Optional) Configure the DN used to authenticate access to the directory.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
  ldap-authentication]
user@host# set bind-dn bind-dn
```

5. (Optional) Configure the password that the SAE uses to authenticate its access to the directory to search for the subscriber profile. If you do not specify a bind DN or bind password, the SAE uses anonymous access.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
  ldap-authentication]
user@host# set bind-password bind-password
```

6. (Optional) Configure the additional LDAP search filter that the SAE uses to search the directory for the subscriber profile.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
  ldap-authentication]
user@host# set search-filter search-filter
```

7. (Optional) Enable the secure protocol used for LDAP connections with the directory. LDAPS, the only secure protocol supported, causes communication with the directory to be encrypted with Secure Sockets Layer (SSL).

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
  ldap-authentication]
user@host# set ldaps
```

8. (Optional) Configure the base DN for searching entries in the directory.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
  ldap-authentication]
user@host# set search-base-dn search-base-dn
```

9. (Optional) Configure the name of the directory attribute that holds the username.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
  ldap-authentication]
user@host# set name-attribute name-attribute
```

10. (Optional) Configure the name of the directory attribute that stores the password.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
 ldap-authentication]
user@host# set password-attribute password-attribute
```

11. (Optional) Configure the name of the directory attribute that contains the name of the service bundle that is used for subscriber authentication. This value is made available to the subscriber classification process and can be used to select the subscriber profile to load.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
 ldap-authentication]
user@host# set service-bundle-attribute service-bundle-attribute
```

12. (Optional) Configure the name of the LDAP attribute that contains the value of the session volume quota. The LDAP plug-in sets the session volume quota to this value.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
 ldap-authentication]
user@host# set session-volume-quota session-volume-quota
```

13. (Optional) Configure the maximum time the SAE waits for a response from a directory server.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
 ldap-authentication]
user@host# set timeout timeout
```

14. (Optional) From operational mode, verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name ldapAuth
 ldap-authentication]
user@host# show
method search;
search-filter (objectClass=umcSubscriber);
name-attribute uniqueId;
timeout 5000;
```

Configuring UDP Ports for RADIUS Plug-Ins

In RADIUS packets that RADIUS plug-ins send to a RADIUS server, the plug-in uses an identifier field to match requests to replies. This field provides for a maximum of 256 identifiers. Once all identifiers are used, the plug-in cannot send any more requests until it receives replies that match the requests already sent. In high-load systems, this limit can slow performance.

To overcome this limitation, you can configure a pool of UDP ports for RADIUS plug-ins. Having a pool of ports allows RADIUS plug-ins to create one queue per port to wait for RADIUS replies. Each queue can wait for 256 RADIUS packets. The RADIUS plug-ins send RADIUS packets through the pool of ports in a round-robin mode.

You can configure a global source UDP port or pool of ports that RADIUS plug-ins use to communicate with RADIUS servers. You can also configure UDP ports for each plug-in instance. If you do not configure a UDP port for a plug-in instance, the plug-in uses the global UDP port.

Use the following configuration statement to configure global configuration ports:

```
shared sae configuration global-radius-udp-port {
    udp-port;
}
```

To configure global UDP ports:

1. From configuration mode, access the global RADIUS UDP port configuration. In this sample procedure, the UDP port is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration
global-radius-udp-port
```

2. Configure the source UDP port or a pool of ports that RADIUS plug-ins use to communicate with RADIUS servers.

```
[edit shared sae group west-region configuration global-radius-udp-port]
user@host# set udp-port
```

- Related Topics**
- Using Flexible RADIUS Packet Definitions on page 127
 - Configuring a RADIUS Packet Template on page 125
 - Creating RADIUS Peers on page 93
 - Configuring UDP Ports for RADIUS Plug-Ins (C-Web Interface)
 - Overview of Flexible RADIUS Plug-Ins on page 118

Defining RADIUS Packets for Flexible RADIUS Plug-Ins

1. Overview of Flexible RADIUS Plug-Ins on page 118
2. Defining the Values of RADIUS Attributes on page 121
3. Configuring a RADIUS Packet Template on page 125
4. Using Flexible RADIUS Packet Definitions on page 127

Overview of Flexible RADIUS Plug-Ins

Flexible RADIUS accounting and authentication plug-ins allow you to define the content of RADIUS packets that the SAE sends to RADIUS servers. You can specify which attributes are included in different types of RADIUS packets (for example, session start or stop requests, or accounting on or off requests). You can also specify what information is contained in the attribute fields.

A RADIUS attribute configuration consists of RADIUS attribute instances. Each instance defines attributes for a specific type of packet—For example, start requests or accounting off requests.

Within each attribute instance, you define individual RADIUS attributes. The following is a RADIUS attribute instance for authentication requests:

```
radius-attributes auth {
  attributes {
    User-Name loginId;
    User-Password password;
    NAS-Identifier localNasId;
    NAS-IP-Address localNasIp;
    NAS-Port nasPort;
  }
}
```

Each RADIUS packet template can consist of multiple RADIUS attribute instances.

Using Default RADIUS Templates

The SRC software comes with two default templates:

- **stdAcct**—Defines RADIUS accounting packets and is used in the default RADIUS flexible accounting plug-in instance `flexRadiusAcct`.
- **stdAuth**—Defines RADIUS authentication packets and is used in the default RADIUS flexible authentication plug-in instance `flexRadiusAuth`.

Naming RADIUS Attribute Instances

Attribute instances define attributes for a specific type of RADIUS packet. The name that you assign to an attribute instance specifies the type of packet to which the attribute definition is applied. Table 11 on page 119 lists the available packet types.

Table 11: RADIUS Attribute Instance Names

Attribute Instance (Packet-Type)	Type of RADIUS Packet to Which Attribute Definition Is Applied
acct	Any accounting request
auth	Any authentication request
authresp	Any authorization response
dhcprsp	DHCP response
off	Accounting-Off requests
on	Accounting-On requests
onoff	Accounting-On or Accounting-Off requests
start	Start requests

Table 11: RADIUS Attribute Instance Names *(continued)*

Attribute Instance (Packet-Type)	Type of RADIUS Packet to Which Attribute Definition Is Applied
startstop	Start, Stop, or Interim Update requests
stop	Stop or Interim Update requests
svcacct	Service Session Start, Stop, or Interim requests
svcrep	Any service authorization response
svcstart	Service Session Start requests
svcstop	Service Session Stop or Interim requests
useracct	Subscriber Session Start, Stop, or Interim requests
userrep	Any subscriber authorization response
userstart	Subscriber Session Start requests
userstop	Subscriber Session Stop, or Interim requests

Defining RADIUS Attributes

RADIUS attribute definitions consist of a RADIUS attribute and a value for the RADIUS attribute.

You can define values for standard RADIUS attributes or JUNOS vendor-specific attributes (VSAs).

Standard RADIUS Attributes

For standard RADIUS attributes, use a name or number as defined in *RFC 2865—Remote Authentication Dial In User Service (RADIUS) (June 2000)*, *RFC 2866—RADIUS Accounting (June 2000)*, or *RFC 2869—RADIUS Extensions (June 2000)*. For a full list, see www.iana.org/assignments/radius-types.

Juniper Networks VSAs

For Juniper Networks VSAs, use one of the following formats:

- Vendor-Specific.4874. <vsa#> . <type>
- 26.4874. <vsa#> . <type>

where <type> is one of the following:

- text—Indicates that the value is 1–253 octets containing UTF-8 encoded characters
- string—Indicates that the value is 1–253 octets containing binary data

- address—Indicates that the value is a 32-bit value
- integer—Indicates that the value is a 32-bit unsigned value
- time—Indicates that the value is a 32-bit unsigned value, seconds since 00:00:00 UTC, January 1, 1970

The following is an example of RADIUS attribute instances that define RADIUS VSAs.

```
radius-attributes svcresp {
  attributes {
    Session-Timeout setSessionTimeout(ATTR);
    Idle-Timeout setIdleTimeout(ATTR);
    vendor-specific.Juniper.Sdx-Session-Volume-Quota setSessionVolumeQuota(ATTR);
    vendor-specific.WISPr.Redirection-URL "setProperty(\"startURL=%s\" % ATTR)";
    vendor-specific.WISPr.Bandwidth-Min-Up "setSubstitution(\"min_up_rate=%s\" % ATTR)";
    vendor-specific.WISPr.Bandwidth-Min-Down "setSubstitution(\"min_down_rate=%s\" % ATTR)";
    vendor-specific.WISPr.Bandwidth-Max-Up "setSubstitution(\"max_up_rate=%s\" % ATTR)";
    vendor-specific.WISPr.Bandwidth-Max-Down "setSubstitution(\"max_down_rate=%s\" % ATTR)";
  }
}
radius-attributes dhcpreap {
  attributes {
    Framed-Pool setPoolName(ATTR);
    Framed-IP-Address setUserIpAddress(ATTR);
    26.4874.1.text setAuthVirtualRouterName(ATTR);
    26.4874.2.text setPoolName(ATTR);
    26.4874.31.text setServiceBundle(ATTR);
  }
}
```

- Related Topics**
- Using Flexible RADIUS Packet Definitions on page 127
 - Configuring a RADIUS Packet Template on page 125
 - Configuring UDP Ports for RADIUS Plug-Ins on page 117
 - Defining the Values of RADIUS Attributes on page 121

Defining the Values of RADIUS Attributes

The values of RADIUS attributes can be a standard value (see Table 12 on page 121) or an expression. Expressions are evaluated with Python. For example: `lowWord(inOctets)` extracts the lower 32 bits of the 64-bit `inOctets` counter. You can define multiple values for an expression in a comma-separated list.

Table 12: Standard Values for RADIUS Attributes

Value	Type of Plug-In	Comments
accountingId	User and service tracking	
authUserId	Service tracking	

Table 12: Standard Values for RADIUS Attributes *(continued)*

Value	Type of Plug-In	Comments
dhcp	User and service tracking	Provides access to DHCP packet. See Table 7 on page 63 for details.
domain	Authorization	
eventTime	User and service tracking	Seconds since 1970-01-01T00:00Z
ifRadiusClass	User and service tracking	
ifSessionId	User and service tracking	
inOctets	Service tracking	64-bit counter
inPackets	Service tracking	
interfaceAlias	User and service tracking	
interfaceDescr	User and service tracking	
interfaceName	User and service tracking	
localNasId	All	Configured NAS-ID
localNasIp	All	Configured NAS-IP
loginId	User and service authorization	ID provided by the subscriber; the loginId value is not separated into UID and domain name.
loginName	User and service tracking	Name that the subscriber uses to log in to portal
nasIp	User and service tracking	NAS IP address of the router
nasPort	User and service tracking	32-bit integer
outOctets	Service tracking	64-bit counter
outPackets	Service tracking	
password	User and service authorization	
portId	User and service tracking	ID of the port on the JUNOSe router; for example, FastEthernet 3/1:2001
primaryUserName	User and service tracking	Name that the subscriber uses for DHCP/PPP authentication

Table 12: Standard Values for RADIUS Attributes *(continued)*

Value	Type of Plug-In	Comments
radiusClass	User tracking, user and service authorization	For service tracking, this value is taken from the RADIUS Access-Accept response. If the response does not contain a value, the RADIUS class defined in the service definition is used. This attribute can be set by an authorization response.
replyMessage	User and service authorization	This attribute can only be set.
routerName	User and service tracking	
serviceBundle	User tracking and authorization	This attribute can be set by an authorization response.
serviceName	Service tracking	Sets an arbitrary attribute (for example, class) to the name of the service.
serviceSessionName	Service tracking	Named service session; empty for default session
serviceSessionTag	Service tracking	
sessionId	User and service tracking	
sessionTime	User and service tracking	
sessionTimeout	User tracking, user and service authorization	This attribute can be set by an authorization response.

Table 12: Standard Values for RADIUS Attributes (continued)

Value	Type of Plug-In	Comments
sessionVolumeQuota	User authorization	<p>This attribute can only be set. It is sent for session tracking events and can be returned by service authorization events. It can be set and retrieved through the portal API and can also be defined through an LDAP attribute in the service definition.</p> <p>If the attribute is defined multiple times, the following precedence is observed:</p> <ol style="list-style-type: none"> 1. Service definition (lowest) 2. Authorization 3. API call (highest) <p>NOTE: The SAE does not enforce a volume quota directly; it only makes the attribute available to an external application that can control the volume quota.</p>
setAcctInterimTime	User authorization	Integer
setAuthVirtualRouterName	DHCP authorization	Text
setIdleTimeout(ATTR)	User authorization	
setLoadServices(ATTR)	User authorization	This attribute can only be set.
setPoolName	DHCP authorization	Text
setRadiusClass(ATTR)	User and service authorization	
setReplyMessage(ATTR)	User and service authorization	
setSessionTimeout(ATTR)	User and service authorization	
setServiceBundle(ATTR)	User authorization	
setSessionVolumeQuota(ATTR)	User authorization	
setSubstitution	User authorization	Text. Substitutions can be set only for service sessions.
setTerminateTime	User authorization	Text
setUserIpAddress	DHCP authorization	Integer

Table 12: Standard Values for RADIUS Attributes *(continued)*

Value	Type of Plug-In	Comments
sspHost	User and service tracking	
terminateCause	User and service tracking	
uid	User and service authorization	
userDn	User and service tracking	
userIpAddress	User and service tracking	
userMacAddress	User and service tracking	
userRadiusClass	Service tracking	RADIUS class of associated subscriber session
userSessionId	Service tracking	RADIUS session ID of associated subscriber session

Related Topics

- Overview of Flexible RADIUS Plug-Ins on page 118
- Using Flexible RADIUS Packet Definitions on page 127
- Configuring a RADIUS Packet Template on page 125
- Configuring UDP Ports for RADIUS Plug-Ins on page 117

Configuring a RADIUS Packet Template

There are two ways to define RADIUS packets for flexible RADIUS accounting and authentication plug-ins:

- Define attributes in a template, and then apply the template to flexible RADIUS accounting and authentication plug-ins.
- Define attributes in the packet definition configuration of a flexible plug-in instance. These definitions override definitions in packet templates.

Use the following configuration statements to configure a RADIUS packet template:

```
shared sae configuration radius-packet-template name ...
shared sae configuration radius-packet-template name radius-attributes name ...
shared sae configuration radius-packet-template name radius-attributes name
  attributes name {
    value ;
  }
shared sae configuration plug-ins name name flex-radius-accounting
  radius-packet-definition name ...
shared sae configuration plug-ins name name flex-radius-accounting
  radius-packet-definition name attributes name {
    value ;
  }
```

```

shared sae configuration plug-ins name name flex-radius-authentication
radius-packet-definition name ...
shared sae configuration plug-ins name name flex-radius-authentication
radius-packet-definition name attributes name {
    value ;
}

```

To configure a template:

1. From configuration mode, access the RADIUS packet template configuration. In this sample procedure, the stdAcct template is configured in the west-region SAE group.

```

user@host# edit shared sae group west-region configuration
radius-packet-template stdAcct

```

2. Create an attribute instance using the names in Table 11 on page 119, and enter the configuration for the RADIUS attribute instance.

```

[edit shared sae group west-region configuration radius-packet-template stdAcct]
user@host# edit radius-attributes name

```

3. Add RADIUS attribute definitions to the attribute instance. Repeat this step for each attribute.

```

[edit shared sae group west-region configuration radius-packet-template stdAcct
radius-attributes svcstop]
user@host# set attributes name value

```

For example:

```

[edit shared sae group west-region configuration radius-packet-template stdAcct
radius-attributes svcstop]
user@host# set attributes Acct-Session-ID sessionId

```

4. (Optional) Verify the configuration of your attribute instance.

```

[edit shared sae group west-region configuration radius-packet-template
stdAcct radius-attributes svcstop]
user@host# show
attributes {
    Acct-Input-Octets lowWord(inOctets);
    Acct-Output-Octets lowWord(outOctets);
    Acct-Input-Packets lowWord(inPackets);
    Acct-Output-Packets lowWord(outPackets);
    Acct-Input-Gigawords highWord(inOctets);
    Acct-Output-Gigawords highWord(outOctets);
}

```

5. (Optional) Verify the configuration of the RADIUS packet template.

```

[edit shared sae group west-region configuration radius-packet-template
stdAcct radius-attributes svcstop]
user@host# up
[edit shared sae group west-region configuration radius-packet-template

```

```

stdAcct]
user@host# show
radius-attributes svcstop {
  attributes {
    Acct-Input-Octets lowWord(inOctets);
    Acct-Output-Octets lowWord(outOctets);
    Acct-Input-Packets lowWord(inPackets);
    Acct-Output-Packets lowWord(outPackets);
    Acct-Input-Gigawords highWord(inOctets);
    Acct-Output-Gigawords highWord(outOctets);
  }
}
radius-attributes stop {
  attributes {
    Acct-Session-Time sessionTime;
    Acct-Terminate-Cause terminateCause;
  }
}
radius-attributes svcacct {
  attributes {
    Class radiusClass;
  }
}
radius-attributes acct {
  attributes {
    Acct-Session-Id sessionId;
    NAS-Identifier localNasId;
    NAS-IP-Address localNasIp;
    Event-Time eventTime;
  }
}
radius-attributes startstop {
  attributes {
    Acct-Multi-Session-Id ifSessionId;
    NAS-Port-Id "\"%s %s\" %(routerName, portId or interfaceName)";
    NAS-Port "nasPort or None";
  }
}

```

- Related Topics**
- Using Flexible RADIUS Packet Definitions on page 127
 - Overview of Flexible RADIUS Plug-Ins on page 118
 - Defining the Values of RADIUS Attributes on page 121

Using Flexible RADIUS Packet Definitions

This topic shows some of the ways you can use flexible RADIUS packet definitions. Remember that the name of the attribute instance determines the type of RADIUS packet in which the packet definition is used.

- To use the Challenge Handshake Authentication Protocol (CHAP) to authenticate subscribers, include the Chap-Password and optionally the Chap-Challenge attributes in authentication requests. (We recommend that you use Chap-Password only. Use Chap-Challenge only if required.) To use a CHAP password, include the following in attribute instance auth:

`Chap-Password = password`

- To cause the Calling-Station-Id attribute to use the subscriber's MAC address:

`Calling-Station-Id = userMacAddress`

- To set the value to prefix N followed by the service name and the prefix S followed by the service session name:

`'N'+serviceName, 'S'+serviceSessionName`

- To construct a value for the Nas-Port-Id attribute by concatenating the value of routerName, a space, and the Nas-Port-ID on the router:

`Nas-Port-Id=routerName + " " + portId`

For example, the constructed value might be:

`default@phoenix FastEthernet 4/2`

- The following example sets the User-Name attribute as follows:
- Sets the value to accountingId, or
- If accountingId is empty, sets the value to loginName, or
- If loginName is also empty, sets the value to NN

`User-Name = accountingId or loginName or " NN"`

- To extract the lower 32 bits of the 64-bit inOctet counter:

`Acct-Input-Octets = lowWord(inOctets)`

- To set the counter fields in the RADIUS packet to the appropriate 32-bit values:

`Acct-Input-Octets = lowWord(inOctets)`
`Acct-Output-Octets = lowWord(outOctets)`
`Acct-Input-Packets = inPackets`
`Acct-Output-Packets = outPackets`

`Acct-Input-Gigawords = highWord(inOctets)`
`Acct-Output-Gigawords = highWord(outOctets)`

- The inOctets and outOctets are 64-bit values and must be split into lower 32-bit (Acct-*-Octets) and upper 32-bit (Acct-*-Gigawords) values.
- The inPacket and outPacket counters are 32-bit values and can be assigned directly.

Setting Values in Authentication Response Packets

You can use some special attribute values to set values in authentication response packets. For example:

- `setRadiusClass(ATTR)`
- `setSessionTimeout(ATTR)`
- `setSessionVolumeQuota(ATTR)`

“Overview of Flexible RADIUS Plug-Ins” on page 118 lists the type of packets (authresp, userresp, or svcresp) in which you can use these values.

When the RADIUS client finds one of these attribute values in an authentication response, it binds ATTR to the current attribute and executes the defined expression. The expression calls one of the available set methods to set the value in the plug-in event.

Below are some examples.

- To set a session timeout:


```
Session-Timeout = setSessionTimeout(ATTR)
```
- To set the RADIUS class:


```
Class = setRadiusClass(ATTR)
```
- To set the service bundle in VSA 31:


```
26.4874.31.text = setServiceBundle(ATTR)
```
- To set the session volume quota:


```
26.4874.50.text = setSessionVolumeQuota(ATTR)
```

Selecting IP Address Pools Using DHCP Response Packets

For DHCP subscribers, you can set up RADIUS authorization plug-ins to return to the router attributes that can be used to select a DHCP address such as framed IP address and pool. You can also set up the name of the virtual router on which the address pool is located and select a fixed address for each subscriber.

- Framed IP address—Selects the pool from which the address is allocated; if the framed IP address is not available, the DHCP server allocates the next available address in the pool; use the `setUserIpAddress` value.
- Framed IP pool—Name of the address pool on the router from which an IP address is assigned; use the `setPoolName` value.
- Virtual router name—Name of the virtual router on which the address pool is located; use the `setAuthVirtualRouterName` value.

You can also select a fixed address for each subscriber. If you identify subscribers by port information (for example, NAS-IP and NAS-Port), the authorization response can select a fixed IP address for each subscriber.



NOTE: Parameters set in the DHCP profile override parameters set by DHCP authorization plug-ins.

-
- Related Topics**
- Configuring UDP Ports for RADIUS Plug-Ins on page 117
 - Configuring a RADIUS Packet Template on page 125
 - Defining the Values of RADIUS Attributes on page 121

Configuring Event Publishers

- Special Types of Event Publishers on page 130
- Configuring Global and Default Retailer Event Publishers on page 131

Special Types of Event Publishers

The SCR CLI lets you configure global and default retailer event publishers. You can also configure service-specific event publishers, retailer-specific event publishers, and virtual router-specific event publishers.

Configuring Service-Specific Event Publishers

In the value-added services definition, you can configure two event publishers for a service:

- Authorization plug-ins—Authenticate subscribers of the service and/or authorize service sessions for this service. These plug-in instances are called before a subscription to this service is activated.
- Tracking plug-ins—Track service sessions of this service. These plug-in instances are called when a service session is started and stopped and during interim updates.

Configuring Retailer-Specific Event Publishers

In the retailer definition, you can configure three event publishers for a retailer:

- Authentication plug-ins—Authenticate subscribers who log in to the domains of the retailer. These plug-in instances are called when a subscriber tries to log in to the SAE through the portal login.

If you do not specify retailer-specific authentication plug-ins, the default retailer authentication plug-ins are called. If you do not specify default retailer authentication plug-ins, subscribers are admitted without authentication.

- Tracking plug-ins—Track sessions of subscribers who log in to the domains of the retailer. These plug-in instances are called after a subscriber session has started and when the session is stopped.
- DHCP authorization plug-ins—Authenticate DHCP address requests for subscribers who log in to the domains of the retailer.

Configuring Virtual Router–Specific Event Publishers

In the virtual router definition, you can configure an interface-tracking plug-in event publisher for a virtual router. These plug-in instances are called when a managed interface is started and stopped. They are called after an interface comes up, when new policies are installed on the interface, and when the interface goes down.

- Related Topics**
- Adding Retailers (SRC CLI) on page 137
 - Adding JUNOS Routers and Virtual Routers with the CLI
 - For more information about JUNOS routing platform and virtual routers, see the *SRC-PE Network Guide*
 - Configuring Global and Default Retailer Event Publishers on page 131

Configuring Global and Default Retailer Event Publishers

Use the following configuration statements to configure global and default retailer event publishers.

```
shared sae configuration plug-ins event-publishers {
  subscriber-authorization subscriber-authorization ;
  default-retailer-authentication default-retailer-authentication ;
  default-retailer-dhcp-authentication default-retailer-dhcp-authentication ;
  dhcp-authorization dhcp-authorization ;
  service-authorization service-authorization ;
  subscription-authorization subscription-authorization ;
  subscriber-tracking subscriber-tracking ;
  service-tracking service-tracking ;
  interface-tracking interface-tracking ;
  embedded-admin-server-authorization embedded-admin-server-authorization ;
}
```

To configure global and default retailer event publishers:

1. From configuration mode, access the event publisher configuration. In this sample procedure, the event publishers are configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins
event-publishers
```

2. Configure plug-ins that authorize subscriber sessions.

```
[edit shared sae group west-region configuration plug-ins event-publishers]
user@host# set subscriber-authorization subscriber-authorization
```

3. Configure plug-ins that authenticate subscribers who are assigned to retailer objects that do not specify an authentication plug-in.

```
[edit shared sae group west-region configuration plug-ins event-publishers]
user@host# set default-retailer-authentication default-retailer-authentication
```

4. Configure plug-ins that authenticate DHCP address requests for subscribers who are assigned to retailer objects that do not specify a DHCP authorization plug-in.

```
[edit shared sae group west-region configuration plug-ins event-publishers]
user@host# set default-retailer-dhcp-authentication
default-retailer-dhcp-authentication
```

5. Configure plug-ins that authorize all DHCP address requests for all DHCP subscribers who log in to a portal.

```
[edit shared sae group west-region configuration plug-ins event-publishers]
user@host# set dhcp-authorization dhcp-authorization
```

6. Configure plug-ins that authorize all service sessions.

```
[edit shared sae group west-region configuration plug-ins event-publishers]
user@host# set service-authorization service-authorization
```

7. Configure plug-ins that authorize subscribers to change their subscriptions.

```
[edit shared sae group west-region configuration plug-ins event-publishers]
user@host# set subscription-authorization subscription-authorization
```

8. Configure plug-ins that collect accounting data for all subscriber sessions.

```
[edit shared sae group west-region configuration plug-ins event-publishers]
user@host# set subscriber-tracking subscriber-tracking
```

9. Configure plug-ins that collect accounting data for all service sessions.

```
[edit shared sae group west-region configuration plug-ins event-publishers]
user@host# set service-tracking service-tracking
```

10. Configure plug-ins, including network information collector (NIC) SAE plug-in agents, that collect accounting data for all interfaces that the SAE manages.

```
[edit shared sae group west-region configuration plug-ins event-publishers]
user@host# set interface-tracking interface-tracking
```

11. Configure plug-ins that authorize administrators to connect to the embedded Web server, which is used to access SAE Web Admin.

```
[edit shared sae group west-region configuration plug-ins event-publishers]
user@host# set embedded-admin-server-authorization
embedded-admin-server-authorization
```

12. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins event-publishers]
user@host# show
subscriber-authorization ;
default-retailer-authentication ldapAuth;
default-retailer-dhcp-authentication ;
dhcp-authorization ;
service-authorization ;
subscription-authorization ;
subscriber-tracking fileAcct;
service-tracking fileAcct;
interface-tracking ;
embedded-admin-server-authorization adminLdap;
```

- Related Topics**
- Special Types of Event Publishers on page 130
 - Configuring an SAE Group

Chapter 8

Configuring Subscribers and Subscriptions (SRC CLI)

- Overview of Configuring Subscribers and Subscriptions on page 135
- Enabling the Subscriber and Subscription Configuration on the SRC CLI on page 136
- Adding Subscribers (SRC CLI) on page 137
- Adding Retailers (SRC CLI) on page 137
- Configuring Administrative Information for Retailers (SRC CLI) on page 139
- Adding Subscriber Folders (SRC CLI) on page 140
- Adding Residential Subscribers (SRC CLI) on page 141
- Configuring Administrative Information for Residential Subscribers (SRC CLI) on page 144
- Adding Enterprises (SRC CLI) on page 145
- Configuring Administrative Information for Enterprise Subscribers (SRC CLI) on page 147
- Adding Sites (SRC CLI) on page 148
- Adding Devices as Subscribers (SRC CLI) on page 149
- Adding Managers (SRC CLI) on page 151
- Configuring Subscriptions (SRC CLI) on page 153
- Configuring Accesses (SRC CLI) on page 155

Overview of Configuring Subscribers and Subscriptions

- Specifying the Activation Order for Subscriptions on page 135
- Inheritance of Properties and Subscriptions on page 136

Specifying the Activation Order for Subscriptions

You can specify the order in which the SAE activates subscriptions that are set up to activate on login for a particular subscriber. To specify the order, you define a precedence for the activation of each subscription. The SAE activates services in ascending order of precedence; if multiple services have the same precedence, the SAE activates them in an unspecified order.

You can configure the activation order by setting the **activation-order** option when you configure a subscription to a service with the SRC CLI. The enterprise manager portal automatically sets the activation order of some subscriptions to ensure they are activated before other subscriptions that depend on them.

Inheritance of Properties and Subscriptions

Subordinate subscribers inherit properties and SAE subscriptions from their parent subscribers, unless you specify a different value for the subordinate. Properties that a subscriber can inherit include the maximum number of concurrent logins and the session timeout. For example, if you configure a subscription to a video service for an enterprise and configure a different subscription to the same video service for a site within that enterprise, the site uses its own subscription rather than the inherited subscription.

- Related Topics**
- Overview of Subscriptions on page 4
 - Overview of Subscribers on page 3
 - Enabling the Subscriber and Subscription Configuration on the SRC CLI on page 136
 - Adding Subscribers (SRC CLI) on page 137
 - Configuring Subscriptions (SRC CLI) on page 153
 - Creating and Tracking Subscriber Sessions on page 86

Enabling the Subscriber and Subscription Configuration on the SRC CLI

Before you can configure subscribers and subscriptions with the SRC CLI, you must enable the policy, service, and subscriber editor on the SRC CLI. To do so:

- In operational mode, enter the following command:

```
user@host> enable component editor
```

If you are using multiple C-series Controllers, we recommend that you enable the policy, service, and subscriber editor on only one C-series Controller on your network. If you enable the editor on multiple platforms, there is a risk that configuration changes will conflict. In this case, the second edit that is committed to the platform is lost.

- Related Topics**
- Enabling the Subscriber and Subscription Configuration on the C-Web Interface
 - Configuring Subscriptions (SRC CLI) on page 153
 - Overview of Configuring Subscribers and Subscriptions on page 135
 - Enterprise Subscriber and Subscription Hierarchy on page 4

Adding Subscribers (SRC CLI)

The tasks to configure subscribers are:

- Adding Retailers (SRC CLI) on page 137
- Configuring Administrative Information for Retailers (SRC CLI) on page 139
- Adding Subscriber Folders (SRC CLI) on page 140

The subscriber hierarchy requires that the objects immediately subordinate to retailers be subscriber folders. You can, however, use subscriber folders subordinate to other subscriber objects to organize groups of subscribers.

- Adding Residential Subscribers (SRC CLI) on page 141
- Adding Enterprises (SRC CLI) on page 145
- Configuring Administrative Information for Enterprise Subscribers (SRC CLI) on page 147
- Adding Sites (SRC CLI) on page 148
- Adding Devices as Subscribers (SRC CLI) on page 149

After you add subscribers, you can add managers and configure subscriptions.

Related Topics

- Adding Managers (SRC CLI) on page 151
- Configuring Subscriptions (SRC CLI) on page 153
- Adding Subscribers (C-Web Interface)
- Creating and Tracking Subscriber Sessions on page 86
- Overview of Subscribers on page 3
- Overview of Configuring Subscribers and Subscriptions on page 135

Adding Retailers (SRC CLI)

If you customize the SRC software for only one Internet service provider (ISP), use the retailer called *default* that is provided in the sample data. If the SRC software will manage multiple ISPs, add a retailer for each ISP.

Use the following configuration statements to add a retailer:

```
subscribers retailer name {
  domain-name [ domain-name... ];
  authentication-plug-in [ authentication-plug-in... ];
  dhcp-authentication-plug-in [ dhcp-authentication-plug-in... ];
  tracking-plug-in [ tracking-plug-in... ];
  maximum-login maximum-login ;
  session-timeout session-timeout ;
  scope [ scope... ];
  substitution [ substitution... ];
}
```

To add a retailer:

1. From configuration mode, enter the retailer configuration. In this procedure, retailer-one is the name of the retailer.

```
user@host# edit subscribers retailer retailer-one
```

2. Configure the domain name(s) associated with the retailer.

```
[edit subscribers retailer retailer-one]
user@host# set domain-name [ domain-name... ]
```

3. (Optional) Configure the plug-in(s) used to authenticate subscribers who log in to the domains specified for this retailer.

```
[edit subscribers retailer retailer-one]
user@host# set authentication-plug-in [ authentication-plug-in... ]
```

4. (Optional) Configure the DHCP authorization plug-in(s) used to authenticate DHCP discover requests for subscribers who log in to the domains specified for this retailer.

```
[edit subscribers retailer retailer-one]
user@host# set dhcp-authentication-plug-in [ dhcp-authentication-plug-in... ]
```

5. (Optional) Configure the plug-in(s) used for accounting or tracking subscriber sessions.

```
[edit subscribers retailer retailer-one]
user@host# set tracking-plug-in [ tracking-plug-in... ]
```

6. (Optional) Configure the maximum number of concurrent logins for subscribers associated with this retailer.

```
[edit subscribers retailer retailer-one]
user@host# set maximum-login maximum-login
```

7. (Optional) Configure the timeout for subscriber sessions.

```
[edit subscribers retailer retailer-one]
user@host# set session-timeout session-timeout
```

8. (Optional) Assign service scopes to the retailer.

```
[edit subscribers retailer retailer-one]
user@host# set scope [ scope... ]
```

9. (Optional) Configure the actual values for parameters associated with this retailer.

```
[edit subscribers retailer retailer-one]
user@host# set substitution [ substitution... ]
```

10. (Optional) Verify your configuration.

```
[edit subscribers retailer retailer-one]
user@host# show
domain-name abc.com;
authentication-plugin flexRadiusAuth;
tracking-plugin fileAcct;
maximum-login 8;
session-timeout 6000;
```

- Related Topics**
- Adding Retailers (C-Web Interface)
 - Configuring Administrative Information for Retailers (SRC CLI) on page 139
 - Adding Managers (SRC CLI) on page 151
 - Overview of Subscriptions on page 4
 - Overview of Subscribers on page 3

Configuring Administrative Information for Retailers (SRC CLI)

Use the following configuration statements to configure administrative information about the retailer:

```
subscribers retailer name info {
  contact contact ;
  e-mail e-mail ;
  url url ;
}
```

To add administrative information about retailers:

1. From configuration mode, enter the retailer subscriber info configuration. In this procedure, retailer-one is the name of the retailer.

```
user@host# edit subscribers retailer retailer-one info
```

2. (Optional) Configure a contact name for the retailer.

```
[edit subscribers retailer retailer-one info]
user@host# set contact contact
```

3. (Optional) Configure an e-mail address for the retailer.

```
[edit subscribers retailer retailer-one info]
user@host# set e-mail e-mail
```

4. (Optional) Configure a URL for the retailer.

```
[edit subscribers retailer retailer-one info]
user@host# set url url
```

5. (Optional) Verify your configuration.

```
[edit subscribers retailer retailer-one info]
user@host# show
contact "Mary Smith";
e-mail msmith@abc.com;
url www.abc.com;
```

- Related Topics**
- Configuring Administrative Information for Retailers (C-Web Interface)
 - Configuring Administrative Information for Enterprise Subscribers (SRC CLI) on page 147
 - Adding Retailers (SRC CLI) on page 137
 - Overview of Subscribers on page 3

Adding Subscriber Folders (SRC CLI)

You can create subscriber folders for retailers, existing subscriber folders, enterprises, and sites. You must create a subscriber folder in a retailer object before you can add other types of subscribers.

Use the following configuration statements to configure subscriber folders:

```
subscribers retailer name subscriber-folder folder-name {
  maximum-login maximum-login ;
  session-timeout session-timeout ;
  scope [ scope.. .];
  substitution [ substitution... ];
}
```

To create a subscriber folder:

1. From configuration mode, enter the subscriber folder configuration. In this procedure, retailer-one is the name of the retailer and local is the name of the subscriber folder.

```
user@host# edit subscribers retailer retailer-one subscriber-folder local
```

2. (Optional) Configure the maximum number of concurrent logins for subscribers associated with this folder.

```
[edit subscribers retailer retailer-one subscriber-folder local]
user@host# set maximum-login maximum-login
```

3. (Optional) Configure the timeout for subscriber sessions associated with this folder.

```
[edit subscribers retailer retailer-one subscriber-folder local]
user@host# set session-timeout session-timeout
```

4. (Optional) Assign service scopes to the folder.

```
[edit subscribers retailer retailer-one subscriber-folder local]
```

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

5. (Optional) Configure the actual values for parameters associated with this folder.

```
[edit subscribers retailer retailer-one subscriber-folder local]
user@host# set substitution [ substitution... ]
```

6. (Optional) Verify your configuration.

```
[edit subscribers retailer retailer-one subscriber-folder local]
user@host# show
session-timeout 9000;
scope POP-Boston;
```

- Related Topics**
- Adding Subscriber Folders (C-Web Interface)
 - Adding Subscribers (SRC CLI) on page 137
 - Adding Retailers (SRC CLI) on page 137
 - Overview of Subscribers on page 3
 - Overview of Configuring Subscribers and Subscriptions on page 135

Adding Residential Subscribers (SRC CLI)

Use the following configuration statements to configure residential subscribers:

```
subscribers retailer name subscriber-folder folder-name subscriber name {
  common-name common-name ;
  surname surname ;
  given-name given-name ;
  initials initials ;
  anonymous;
  ip-address ip-address ;
  interface-name interface-name ;
  maximum-login-group maximum-login-group ;
  display-name display-name ;
  encrypted-password encrypted-password ;
  plain-text-password;
  maximum-login maximum-login ;
  session-timeout session-timeout ;
  accounting-user-id accounting-user-id ;
  substitution [ substitution... ];
}
```

To add a residential subscriber:

1. From configuration mode, enter the residential subscriber configuration. In this procedure, peter is the name of the subscriber record.

```
user@host# edit subscribers retailer default subscriber-folder local subscriber
peter
```

2. Configure the name that defines the subscriber in the directory.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set common-name common-name
```

3. Configure the subscriber's last name.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set surname surname
```

4. (Optional) Configure the subscriber's first name.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set given-name given-name
```

5. (Optional) Configure the subscriber's middle initial(s)

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set initials initials
```

6. (Optional) Specify whether the subscriber profile created with this subscriber definition is a shared profile. Subscribers cannot modify shared profiles.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set anonymous
```

7. (Optional) Configure the IP address for subscribers who have fixed IP addresses, and for whom the SRC does not learn addresses through its management of routers or through calls to its notification API.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set ip-address ip-address
```

8. (Optional) Configure the type and specifier of the router interface and virtual router that manage this subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set interface-name interface-name
```

9. (Optional) Configure the maximum number of concurrent logins for this subscriber and all subordinate objects.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set maximum-login-group maximum-login-group
```

10. (Optional) Configure the subscriber's name as it appears in login screens.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set display-name display-name
```

11. (Optional) Configure the login password and type of encryption.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
```

```
user@host# set encrypted-password encrypted-password
```

12. (Optional) Configure the plain text password.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set plain-text-password
```

13. (Optional) Configure the maximum number of concurrent logins for subscribers associated with this subscriber definition.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set maximum-login maximum-login
```

14. (Optional) Configure the timeout for subscriber sessions associated with this subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set session-timeout session-timeout
```

15. (Optional) Configure the value that identifies the subscriber in accounting records; for a household subscriber, all subordinate subscribers generally use the same ID.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set accounting-user-id accounting-user-id
```

16. (Optional) Assign service scopes to the subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set scope [ scope... ]
```

17. (Optional) Configure the actual values for parameters associated with this subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter]
user@host# set substitution [ substitution... ]
```

18. (Optional) Verify your configuration.

```
[edit subscribers retailer default subscriber-folder local subscriber
peter]
user@host# show
common-name psmith;
surname smith;
initials A;
anonymous;
ip-address 10.10.62.3;
interface-name fastethernet6/0.1@vrName@routerName;
encrypted-password abcdefh;
session-timeout 9000;
```

- Related Topics**
- Adding Residential Subscribers (C-Web Interface)
 - Overview of Subscribers on page 3
 - Overview of Configuring Subscribers and Subscriptions on page 135
 - Residential Subscriber Login and Processes on page 11

Configuring Administrative Information for Residential Subscribers (SRC CLI)

Use the following configuration statements to configure administrative information about the subscriber:

```
subscribers retailer name subscriber-folder folder-name subscriber name info {
    home-phone home-phone ;
    additional-phone additional-phone ;
    fax fax ;
    e-mail e-mail ;
    city city ;
    street street ;
    postal-code postal-code ;
    language language ;
    job job ;
    description description ;
}
```

To add administrative information about residential subscribers:

1. From configuration mode, enter the residential subscriber info configuration. In this procedure, peter is the name of the subscriber.

```
user@host# edit subscribers retailer default subscriber-folder local subscriber peter info
```

2. (Optional) Configure a home phone number for the subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter info]
user@host# set home-phone home-phone
```

3. (Optional) Configure a second phone number for the subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter info]
user@host# set additional-phone additional-phone
```

4. (Optional) Configure a fax number for the subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter info]
user@host# set fax fax
```

5. (Optional) Configure an e-mail address for the subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter info]
user@host# set e-mail e-mail
```


6. (Optional) Configure the city for the subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter info]
user@host# set city city
```

7. (Optional) Configure the street address for the subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter info]
user@host# set street street
```

8. (Optional) Configure the postal code for the subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter info]
user@host# set postal-code postal-code
```

9. (Optional) Configure the language of the subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter info]
user@host# set language language
```

10. (Optional) Configure the job description of the subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter info]
user@host# set job job
```

11. (Optional) Configure a description for the subscriber.

```
[edit subscribers retailer default subscriber-folder local subscriber peter info]
user@host# set description description
```

Adding Enterprises (SRC CLI)

Use the following configuration statements to add an enterprise subscriber:

```
subscribers retailer name subscriber-folder folder-name enterprise name {
  display-name display-name ;
  accounting-user-id accounting-user-id ;
  description description ;
  scope [ scope... ];
  substitution [ substitution... ];
}
```

To add an enterprise subscriber:

1. From configuration mode, enter the enterprise subscriber configuration. In this procedure, ABCInc is the name of the enterprise subscriber.

```
user@host# edit subscribers retailer default subscriber-folder local enterprise
ABCInc
```

2. (Optional) Configure the name that is displayed in enterprise management portals, if different from the enterprise name.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc]
user@host# set display-name display-name
```

3. (Optional) Configure the name that identifies the enterprise in accounting records.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc]
user@host# set accounting-user-id accounting-user-id
```

4. (Optional) Enter a description of the enterprise.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc]
user@host# set description description
```

5. (Optional) Assign service scopes to the enterprise.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc]
user@host# set scope [ scope... ]
```

6. (Optional) Configure the actual values for parameters associated with this enterprise.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc]
user@host# set substitution [ substitution... ]
```

7. (Optional) Verify your configuration.

```
[edit subscribers retailer default subscriber-folder local enterprise
ABCInc]
user@host# show
display-name ABCInc;
description "This enterprise is sample data for use with JUNOS routers.
```

The attached EntJunose scope contains enterprise services that are designed to work with JUNOS.

```
scope [ EntJunose POP-Ottawa POP-Boca POP-Boston POP-Montreal ];
substitution [ "acct : network = 208.93.36.80 / 28" "eng : network =
208.93.36.64 / 28" ];
```

8. Configure an access subscription for the enterprise. (See “Configuring Accesses (SRC CLI)” on page 155.)

- Related Topics**
- Adding Enterprises (C-Web Interface)
 - Configuring Administrative Information for Enterprise Subscribers (SRC CLI) on page 147
 - Enterprise Subscriber and Subscription Hierarchy on page 4
 - Enterprise Subscriber Login Process on page 26
 - Automatic Activation at Login on page 32

Configuring Administrative Information for Enterprise Subscribers (SRC CLI)

Use the following configuration statements to configure administrative information about the enterprise subscriber:

```
subscribers retailer name subscriber-folder folder-name enterprise name info {
    phone phone ;
    fax fax ;
    po-box po-box ;
    city city ;
    street street ;
    state state ;
    postal-code postal-code ;
}
```

To add administrative information about enterprise subscribers:

1. From configuration mode, enter the enterprise subscriber info configuration. For example:

```
user@host# edit subscribers retailer default subscriber-folder local enterprise  
ABCInc info
```

2. (Optional) Configure a phone number for the subscriber.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc info]  
user@host# set phone phone
```

3. (Optional) Configure a fax number for the subscriber.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc info]  
user@host# set fax fax
```

4. (Optional) Configure a post office box for the subscriber.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc info]  
user@host# set po-box po-box
```

5. (Optional) Configure the city for the subscriber.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc info]  
user@host# set city city
```

6. (Optional) Configure the street address for the subscriber.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc info]  
user@host# set street street
```

7. (Optional) Configure a state for the subscriber.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc info]  
user@host# set state state
```

8. (Optional) Configure the postal code for the subscriber.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc info]
user@host# set postal-code postal-code
```

- Related Topics**
- Configuring Administrative Information for Enterprise Subscribers (C-Web Interface)
 - Adding Enterprises (SRC CLI) on page 145
 - Configuring Administrative Information for Retailers (SRC CLI) on page 139
 - Overview of Subscribers on page 3
 - Enterprise Subscriber Login Process on page 26

Adding Sites (SRC CLI)

Use the following configuration statements to add a site:

```
subscribers retailer name subscriber-folder folder-name enterprise name site
  name {
    network [ network... ];
    display-name display-name ;
    accounting-user-id accounting-user-id ;
    description description ;
  }
```

To add a site:

1. From configuration mode, enter the site configuration. In this procedure, ABCInc is the name of the enterprise, and Montreal is the name of the site.

```
user@host# edit subscribers retailer default subscriber-folder local enterprise
ABCInc site Montreal
```

2. (Optional) Record networks used at the site. If you build a custom enterprise manager application, you can access this information through the enterprise portal APIs.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc site
Montreal]
user@host# set network [ network... ]
```

3. (Optional) Configure the name that is displayed in enterprise management portals, if different from the site name.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc site
Montreal]
user@host# set display-name display-name
```

4. (Optional) Configure the name that identifies the site in accounting records.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc site
Montreal]
user@host# set accounting-user-id accounting-user-id
```

5. (Optional) Enter a description of the site.

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc site
Montreal]
user@host# set description description
```

6. (Optional) Verify your configuration.

```
[edit subscribers retailer default subscriber-folder local enterprise
ABCInc site Montreal]
user@host# show
display-name "Montreal Office of ABC, Inc.";
accounting-user-id abcInc;
description "This enterprise is sample data for use with JUNOS routers.";
```

7. Configure an access for the site. (See “Configuring Accesses (SRC CLI)” on page 155.)

- Related Topics**
- Adding Sites (C-Web Interface)
 - Adding Retailers (SRC CLI) on page 137
 - Adding Subscriber Folders (SRC CLI) on page 140
 - Overview of Configuring Subscribers and Subscriptions on page 135

Adding Devices as Subscribers (SRC CLI)

Configure a device subscriber for subscriber sessions that manage the forwarding interface on JUNOS routing platforms and the router pseudo-subscriber on JUNOS routers.

You can add devices as subscribers to subscriber folders, enterprises, and sites. Use the following configuration statements to add a device as a subscriber:

```
subscribers retailer name subscriber-folder folder-name device device-name {
  display-name display-name ;
  maximum-login maximum-login ;
  accounting-user-id accounting-user-id ;
  substitution [ substitution... ];
}
subscribers retailer name subscriber-folder folder-name enterprise name device
device-name {
  display-name display-name ;
  maximum-login maximum-login ;
  accounting-user-id accounting-user-id ;
  substitution [ substitution.. .];
}
subscribers retailer name subscriber-folder folder-name enterprise name site
name device device-name {
```

```

display-name display-name ;
maximum-login maximum-login ;
accounting-user-id accounting-user-id ;
substitution [ substitution... ];
}

```

To add a device as a subscriber:

1. From configuration mode, enter the device subscriber configuration. In this procedure, default@TMJunosA is the name of the device.

```

user@host# edit subscribers retailer SP-TM subscriber-folder devices device
default@TMJunosA

```

2. (Optional) Configure the name of the device as you want it to appear in SRC applications, such as portals.

```

[edit subscribers retailer SP-TM subscriber-folder devices device
 default@TMJunosA]
user@host# set display-name display-name

```

3. (Optional) Configure the maximum number of concurrent logins for subscribers associated with this device.

```

[edit subscribers retailer SP-TM subscriber-folder devices device
 default@TMJunosA]
user@host# set maximum-login maximum-login

```

4. (Optional) Configure the name that identifies the device in accounting records.

```

[edit subscribers retailer SP-TM subscriber-folder devices device
 default@TMJunosA]
user@host# set accounting-user-id accounting-user-id

```

5. (Optional) Configure the actual values for parameters associated with this device.

```

[edit subscribers retailer SP-TM subscriber-folder devices device
 default@TMJunosA]
user@host# set substitution [ substitution... ]

```

6. (Optional) Verify your configuration.

```

[edit subscribers retailer SP-TM subscriber-folder devices device
 default@TMJunosA]
user@host# show
display-name "Profile for JUNOS router";
accounting-user-id JunosRouter

```

- Related Topics**
- Adding Devices as Subscribers (C-Web Interface)
 - Adding Subscriber Folders (SRC CLI) on page 140
 - Adding Enterprises (SRC CLI) on page 145
 - Overview of Configuring Subscribers and Subscriptions on page 135

Adding Managers (SRC CLI)

Use the following configuration statements to configure a manager:

```

subscribers retailer name manager name {
    role [(administrator | subscription | substitution | activation | vpn)...];
    encrypted-password encrypted-password ;
    plain-text-password;
    description description ;
}
subscribers retailer name subscriber-folder folder-name manager name {
    role [(administrator | subscription | substitution | activation | vpn)...];
    encrypted-password encrypted-password ;
    plain-text-password;
    description description ;
}
subscribers retailer name subscriber-folder folder-name enterprise name manager
name {
    role [(administrator | subscription | substitution | activation | vpn)...];
    encrypted-password encrypted-password ;
    plain-text-password;
    description description ;
}
subscribers retailer name subscriber-folder folder-name enterprise name site
name manager name {
    role [(administrator | subscription | substitution | activation | vpn)...];
    encrypted-password encrypted-password ;
    plain-text-password;
    description description ;
}
subscribers retailer name subscriber-folder folder-name enterprise name access
name manager name {
    role [(administrator | subscription | substitution | activation | vpn)...];
    encrypted-password encrypted-password ;
    plain-text-password;
    description description ;
}
subscribers retailer name subscriber-folder folder-name enterprise name site
name access name manager name {
    role [(administrator | subscription | substitution | activation | vpn)...];
    encrypted-password encrypted-password ;
    plain-text- password ;
    description description ;
}
subscribers retailer name subscriber-folder folder-name device device-name
manager name {
    role [(administrator | subscription | substitution | activation | vpn)...];
    encrypted-password encrypted-password ;
    plain-text-password;
    description description ;
}
subscribers retailer name subscriber-folder folder-name enterprise name device
device-name manager name {
    role [(administrator | subscription | substitution | activation | vpn)...];
    encrypted-password encrypted-password ;

```

```

plain-text-password;
description description ;
}
subscribers retailer name subscriber-folder folder-name enterprise name site
name device device-name manager name {
role [(administrator | subscription | substitution | activation | vpn)...];
encrypted-password encrypted-password ;
plain-text-password;
description description ;
}

```

To add a manager:

1. From configuration mode, enter the manager configuration. In this procedure, we are creating a manager called abcmgr in the ABCInc enterprise.

```

user@host# edit subscribers retailer default subscriber-folder local enterprise
ABCInc manager abcmgr

```

2. (Optional) Configure the privilege level (role) for the manager.

```

[edit subscribers retailer default subscriber-folder local enterprise ABCInc manager
abcmgr]
user@host# set role [(administrator | subscription | substitution | activation |
vpn)...]

```

3. (Optional) Configure an encrypted password for the manager:

```

[edit subscribers retailer default subscriber-folder local enterprise ABCInc manager
abcmgr]
user@host# set encrypted-password encrypted-password

```

4. (Optional) Configure a plain text password for the manager.

```

[edit subscribers retailer default subscriber-folder local enterprise ABCInc manager
abcmgr]
user@host# set plain-text-password plain-text-password

```

5. (Optional) Enter a description for the manager.

```

[edit subscribers retailer default subscriber-folder local enterprise ABCInc manager
abcmgr]
user@host# set description description

```

6. (Optional) Verify your configuration.

```

[edit subscribers retailer default subscriber-folder local enterprise
ABCInc manager abcmgr]
user@host# show
role administrator;
encrypted-password secret;

```

Related Topics ■ Adding Managers (C-Web Interface)

- Overview of Managers on page 6
- Overview of Configuring Subscribers and Subscriptions on page 135

Configuring Subscriptions (SRC CLI)

After you add subscribers, you configure subscriptions for the subscribers. Residential or enterprise subscribers may also be able to configure subscriptions through the portal, and managers assigned to a subscriber object may be able to configure subscriptions for that object.

You must add a service to the directory before you can specify that service for subscribers. See Overview of Services for the SRC Software.

After you configure a subscription to a service, the service is available to the subscriber through the portal. Depending on the configuration, the subscriber may need to activate the service. You can configure schedules to define when services are available to subscribers. See Overview of Service Schedules.

To allow a subscriber to have a number of subscriptions to a service at the same time, each subscription:

- Must have its own parameter substitutions.
- Can be activated or deactivated independently.

An object for each subscription is created in the directory. The name of the object has the following format:

`<ServiceName>%<SubscriptionId>`

- `<ServiceName>` —Name of the service
- `<SubscriptionId>` —Name of the subscription

Other than the naming convention, multiple subscriptions are identical to regular subscriptions.

```
subscribers retailer name subscription subscription-name {
    status (active | suspended | hidden);
    activation (manual | automatically-on-login);
    activation-order activation-order ;
    substitution [ substitution... ];
}
subscribers retailer name subscriber-folder folder-name subscription
subscription-name {
    status (active | suspended | hidden);
    activation (manual | automatically-on-login);
    activation-order activation-order ;
    substitution [ substitution... ];
}
```

```

subscribers retailer name subscriber-folder folder-name subscriber name
  subscription subscription-name {
    status (active | suspended | hidden);
    activation (manual | automatically-on-login);
    activation-order activation-order ;
    substitution [ substitution... ];
  }
subscribers retailer name subscriber-folder folder-name enterprise name
  subscription subscription-name {
    status (active | suspended | hidden);
    activation (manual | automatically-on-login);
    activation-order activation-order ;
    substitution [ substitution... ];
  }
subscribers retailer name subscriber-folder folder-name enterprise name site
name subscription subscription-name {
  status (active | suspended | hidden);
  activation (manual | automatically-on-login);
  activation-order activation-order ;
  substitution [ substitution... ];
}
subscribers retailer name subscriber-folder folder-name enterprise name access
name subscription subscription-name {
  status (active | suspended | hidden);
  activation (manual | automatically-on-login);
  activation-order activation-order ;
  substitution [ substitution... ];
}
subscribers retailer name subscriber-folder folder-name device device-name
  subscription subscription-name {
    status (active | suspended | hidden);
    activation (manual | automatically-on-login);
    activation-order activation-order ;
    substitution [ substitution... ];
  }
subscribers retailer name subscriber-folder folder-name enterprise name device
device-name subscription subscription-name {
  status (active | suspended | hidden);
  activation (manual | automatically-on-login);
  activation-order activation-order ;
  substitution [ substitution... ];
}
subscribers retailer name subscriber-folder folder-name enterprise name site
name device device-name subscription subscription-name {
  status (active | suspended | hidden);
  activation (manual | automatically-on-login);
  activation-order activation-order ;
  substitution [ substitution... ];
}

```

To configure a subscription to a service:

1. From configuration mode, enter the subscription configuration. In this procedure, peter is the name of the subscriber and Video-Gold is the name of the subscription.

```
user@host# edit subscribers retailer default subscriber-folder local subscriber
peter subscription Video-Gold
```

2. (Optional) Configure the status of the service subscription.

```
[edit subscribers retailer default subscriber-folder local subscriber peter
subscription Video-Gold]
user@host# set status (active | suspended | hidden)
```

3. (Optional) Specify how the service is activated.

```
[edit subscribers retailer default subscriber-folder local subscriber peter
subscription Video-Gold]
user@host# set activation (manual | automatically-on-login)
```

4. (Optional) Specify when the SAE should activate this subscription relative to the subscriber's other subscriptions that are configured to activate on login.

```
[edit subscribers retailer default subscriber-folder local subscriber peter
subscription Video-Gold]
user@host# set activation-order activation-order
```

5. (Optional) Configure the actual values for parameters associated with this subscription.

```
[edit subscribers retailer default subscriber-folder local subscriber peter
subscription Video-Gold]
user@host# set substitution [ substitution... ]
```

6. (Optional) Verify your configuration.

```
[edit subscribers retailer default subscriber-folder local subscriber
peter subscription Video-Gold]
user@host# show
status active;
activation manual;
```

- Related Topics**
- Configuring Subscriptions (C-Web Interface)
 - Enabling the Subscriber and Subscription Configuration on the SRC CLI on page 136
 - Overview of Subscriptions on page 4
 - Overview of Configuring Subscribers and Subscriptions on page 135
 - Enterprise Subscriber and Subscription Hierarchy on page 4

Configuring Accesses (SRC CLI)

You must configure an access for an enterprise or a site. An access determines the way that the enterprise or site accesses Internet services, and specifies a set of services that are available to the particular access.

Subscriber classification scripts can use access subscription properties to match the interface in the network with an access in the directory. Typically, the interface alias, interface description, interface name, unique ID, NAS port ID, and router name are used to match an interface to an access.

You can specify multiple accesses; for example, you might want to specify primary and secondary services for Internet access.

```

subscribers retailer name subscriber-folder folder-name enterprise name access
  name {
    routing-protocol routing-protocol ;
    interface-alias interface-alias ;
    interface-description interface-description ;
    interface-name interface-name ;
    unique-id unique-id ;
    port-id port-id ;
    device-name device-name ;
    display-name display-name ;
    accounting-user-id accounting-user-id ;
    substitution [ substitution... ];
  }
subscribers retailer name subscriber-folder folder-name enterprise name site
  name access name {
    routing-protocol routing-protocol ;
    interface-alias interface-alias ;
    interface-description interface-description ;
    interface-name interface-name ;
    unique-id unique-id ;
    port-id port-id ;
    device-name device-name ;
    display-name display-name ;
    accounting-user-id accounting-user-id ;
    substitution [ substitution... ];
  }

```

To configure a subscription to an access service:

1. From configuration mode, enter the subscription configuration. In this procedure, Acme is the name of the enterprise and AcmeAccess is the name of the access.

```

user@host# edit subscribers retailer SP-TM subscriber-folder subscribers
enterprise Acme access AcmeAccess

```

2. (Optional) Record routing protocols used at the enterprise or site. If you build a custom enterprise manager application, you can access this information through the enterprise portal APIs.

```

[edit subscribers retailer SP-TM subscriber-folder subscribers enterprise Acme
access AcmeAccess]
user@host# set routing-protocol routing-protocol

```

3. (Optional) Configure the description of a router interface.

```

[edit subscribers retailer SP-TM subscriber-folder subscribers enterprise Acme
access AcmeAccess]

```

```
user@host# set interface-alias interface-alias
```

4. (Optional) Configure the alternate name of the interface that SNMP uses.

```
[edit subscribers retailer SP-TM subscriber-folder subscribers enterprise Acme
access AcmeAccess]
```

```
user@host# set interface-description interface-description
```

5. (Optional) Configure the name of the interface using your router CLI syntax

```
[edit subscribers retailer SP-TM subscriber-folder subscribers enterprise Acme
access AcmeAccess]
```

```
user@host# set interface-name interface-name
```

6. (Optional) Configure the router's unique ID, which is the index of the router in the SNMP table for all interfaces.

```
[edit subscribers retailer SP-TM subscriber-folder subscribers enterprise Acme
access AcmeAccess]
```

```
user@host# set unique-id unique-id
```

7. (Optional) Configure the network access server (NAS) port ID reported by the JUNOS router through the Common Open Policy Service (COPS).

```
[edit subscribers retailer SP-TM subscriber-folder subscribers enterprise Acme
access AcmeAccess]
```

```
user@host# set port-id port-id
```

8. (Optional) Configure the name of the router to which this access connects.

```
[edit subscribers retailer SP-TM subscriber-folder subscribers enterprise Acme
access AcmeAccess]
```

```
user@host# set router-name router-name
```

9. (Optional) Configure the name that is displayed in enterprise management portals, if different from the service name.

```
[edit subscribers retailer SP-TM subscriber-folder subscribers enterprise Acme
access AcmeAccess]
```

```
user@host# set display-name display-name
```

10. (Optional) Configure the value that identifies the service in accounting records.

```
[edit subscribers retailer SP-TM subscriber-folder subscribers enterprise Acme
access AcmeAccess]
```

```
user@host# set accounting-user-id accounting-user-id
```

11. (Optional) Configure the actual values for parameters associated with this subscription.

```
[edit subscribers retailer SP-TM subscriber-folder subscribers enterprise Acme
access AcmeAccess]
```

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

12. (Optional) Verify your configuration.

```
[edit subscribers retailer SP-TM subscriber-folder subscribers enterprise  
Acme access AcmeAccess]  
user@host# show  
interface-alias cust123-456;  
interface-name fastethernet6/0.1;
```

- Related Topics**
- Configuring Accesses (C-Web Interface)
 - Adding Enterprises (SRC CLI) on page 145
 - Adding Sites (SRC CLI) on page 148
 - Overview of Configuring Subscribers and Subscriptions on page 135
 - Overview of Configuring Classification Scripts on page 46

Part 2

Redirecting Subscriber Traffic Through Redirect Server

- Redirecting Subscriber Traffic on page 161
- Configuring Traffic Redirection (SRC CLI) on page 165

Chapter 9

Redirecting Subscriber Traffic

- Overview of Traffic Redirection on page 161
- Redirect Server Redundancy on page 163

Overview of Traffic Redirection

The redirect server is part of a captive portal system that redirects subscribers' Web requests to a captive portal page. You can use a captive portal page as the initial page a subscriber sees after logging in to a subscriber session and as a page used to receive and manage HTTP requests to unauthorized Web resources.

Proxy Request Management

The redirect server examines requested paths and detects proxy HTTP requests by the proxy prefix “ < scheme > :” followed by the address of the requested host. If the requested URL is served by the captive portal server:

1. The redirect server opens a TCP connection to the captive portal and forwards the request for the URL. The redirect server adds to the request an X-Forwarded-For header that specifies the IP address of the client.
2. The captive portal server inspects the incoming request for the X-Forwarded-For header for the IP address. The captive portal server uses this address instead of the source IP address to determine the originator of the request.
3. If the captive portal authorizes the client and activates a service that enables a direct connection between the client and the proxy, the redirect server then sends the returned data to the subscriber's Web browser.

or

If the requested URL is not served by the captive portal server, the redirect server opens a TCP port (8800 by default) and sends the type of response configured to a subscriber's browser in response to a captured request:

- HTTP 200 OK response with an HTML document that includes the < HTTP-Equiv = "Refresh" > header (default)
- HTTP 302 Found response to a subscriber's browser in response to a captured request

The subscriber browser follows the redirect request, and the proxied request is served by the redirect server again, which opens a connection to the captive portal.

Support for HTTP proxy requests requires the following:

- A local HTTP proxy server that can handle the traffic from all clients configured with a proxy.
- A location for the local HTTP proxy server that is one IP hop from each access router.
- A proxy service that the captive portal server can activate to send proxy requests to the local HTTP proxy server when the portal server authorizes proxy clients.
- A proxy service activation policy that includes a next-hop policy that points to the local HTTP proxy server, and a classifier that matches the client's IP address and the address of the proxy server configured on the client.

Services that the client accesses through the proxy server, such as HTTP and FTP, cannot be activated based on destination address.

You must redirect all ports to the redirect server because you cannot know which ports are configured on the client for the proxy. Consequently, the redirect server receives non-HTTP requests as well as HTTP requests. The non-HTTP requests generate error log entries. To reduce overhead, HTTP error messages are logged as system log debug messages.

HTTP Proxy and DNS

Make sure that your network includes a domain name service (DNS) server to resolve unknown names to a fixed IP address. A DNS server is required because proxy servers can be configured with DNS names in private domains that are not valid in the public environment. You can use the DNS server included with the redirect server, or another DNS server on your network.

The DNS server can be configured on a client with DHCP. Alternatively, the service provider can set up a transparent DNS proxy by configuring a next-hop policy on the JUNOSe router for UDP and TCP port 53 traffic. The policy redirects traffic on these two ports to the redirect server's DNS server.

Because proxy addresses must be resolved even if general access to the Internet is enabled, the DNS server must continue to resolve all client requests for proxy clients. Nonproxy clients can use their regular DNS server after the initial service has been activated.

The redirect server's DNS server either forwards the request to a set of configured DNS servers or resolves the request by using the root domain name server. If a request for an IPv4 address cannot be resolved and the request results in an NXDOMAIN error, the DNS server returns a configurable IP address. The redirect server returns an error message to the clients for any other type of request that cannot be resolved.

Protection Against Denial-of-Service Attacks

The redirect server incorporates a number of properties to protect against denial-of-service attacks. The following list shows the default values set for these properties:

- The redirect server can serve no more than 12,000 requests per minute, with a burst of 18,000 requests.
- The redirect server can serve no more than 25 requests per client per minute, with a burst of 50 requests.
- Incoming requests can be no larger than 4 KB.
- Incoming requests have a time limit of 2 seconds.

You can change the values for any of these properties.

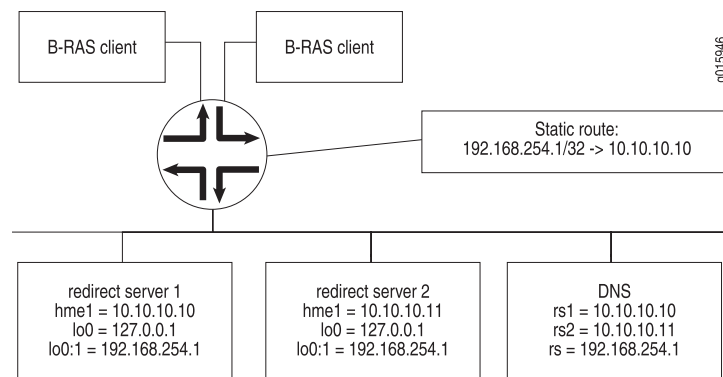
- Related Topics**
- Redirect Server Redundancy on page 163
 - Configuration Statements for the Redirect Server (SRC CLI) on page 165
 - Before You Configure Redundancy for a Redirect Server on page 176
 - Configuring the Redirect Server (SRC CLI) on page 167
 - Configuring the Redirect Server (C-Web Interface)

Redirect Server Redundancy

You can configure the redirect server to provide redundancy to help ensure that a redirect server is always available. You install the redirect server software on two different hosts; then you configure one redirect server as the primary redirect server, and the other as the redundant redirect server. The active and redundant redirect servers regularly poll each other to confirm each other's availability. If the primary redirect server becomes unavailable, the redundant server assumes the active role.

When a redirect server assumes the primary role, it configures on the router a static route from the virtual IP address to the server's real IP address. Clients send requests to the virtual IP address, and the router automatically sends the request to the active redirect server through a static route. The virtual IP address is used only in the static route configured on the router and the next-hop policy installed by SAE. End users do not see the virtual IP address.

Figure 21 on page 164 shows a configuration in which two redirect servers use the same virtual IP address, 192.168.254.1.

Figure 21: Failover of a Redirect Server

- Related Topics**
- Overview of Traffic Redirection on page 161
 - Before You Configure Redundancy for a Redirect Server on page 176
 - Configuring a Redundant Redirect Server (SRC CLI) on page 176
 - Configuring a Redundant Redirect Server (C-Web Interface)

Chapter 10

Configuring Traffic Redirection (SRC CLI)

- Configuration Statements for the Redirect Server (SRC CLI) on page 165
- Before You Configure the Redirect Server on a C-Series Controller on page 166
- Configuring the Redirect Server (SRC CLI) on page 167
- Configuring General Properties for the Redirect Server (SRC CLI) on page 168
- Configuring a Connection Between the Redirect Server and the Directory (SRC CLI) on page 169
- Defining Traffic to Transmit to the Redirect Server (SRC CLI) on page 170
- Changing the Number of Requests That the Redirect Server Accepts (SRC CLI) on page 171
- Specifying Extensions for Files That the Redirect Server Accepts (SRC CLI) on page 172
- Verifying Configuration for the Redirect Server (SRC CLI) on page 173
- Enabling the Redirect Server on page 173
- Configuring the DNS Server for the Redirect Server (SRC CLI) on page 174
- Configuring the Redirect Server to Support HTTP Proxies (SRC CLI) on page 175
- Before You Configure Redundancy for a Redirect Server on page 176
- Configuring a Redundant Redirect Server (SRC CLI) on page 176
- Configuring Logging for the Redirect Server on page 178
- Changing the Configuration for the Redirect Server on page 178
- Assessing Load for Redirect Server (C-Web Interface) on page 178

Configuration Statements for the Redirect Server (SRC CLI)

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

```
redirect-server {  
    tcp-port tcp-port;  
    destination-url destination-url;  
    proxy-support;  
    proxy-destination-url proxy-destination-url;  
    refresh;  
    request-rate request-rate;  
    request-burst-size request-burst-size;
```

```

client-rate client-rate;
client-burst-size client-burst-size;
check-file-extensions;
file-extensions file-extensions;
redundancy;
}
redirect-server ip-redirect{
    interface interface;
    port port;
}
redirect-server ldap {
    url url;
    bind-dn bind-dn;
    bind-password bind-password;
    base-dn base-dn;
}
redirect-server dns {
    enable;
    tcp-port tcp-port;
    udp-port udp-port;
    forwarder forwarder;
    error-ip-address error-ip-address;
}
redirect-server monitor {
    redundant-host-ip-address redundant-host-ip-address;
    virtual-ip-address virtual-ip-address;
    real-ip-address real-ip-address;
    primary-server;
    check-interval check-interval;
    virtual-routers virtual-routers;
}

```

- Related Topics**
- Overview of Traffic Redirection on page 161
 - Configuring the Redirect Server (SRC CLI) on page 167
 - For detailed information about each configuration statement, see the *SRC-PE CLI Command Reference*

Before You Configure the Redirect Server on a C-Series Controller

Before you configure the redirect server on a C-series Controller:

- Configure the connection between the redirect server and the JUNOSe router by configuring policies on the C-series controller:
 - Configure and enable the HTTP local server on the JUNOSe router
 - On the C-series Controller, configure a policy that includes the following policy actions to define which traffic to send to the redirect server:
 - An exception action to specify that an HTTP application receive traffic.
 - An http redirect policy action to specify the URL to receive packets identified in the exception application action.



NOTE: Alternatively, if the distance between the JUNOS routers and the C-series Controller is one hop away, you can configure a next-hop policy on the JUNOS router that specifies a destination address that is the virtual IP address of the active redirect server rather than configuring an SRC policy.

- If you plan to configure a redundant redirect server, make sure that you are familiar with the network configuration required.

Related Topics

- Before You Configure Redundancy for a Redirect Server on page 176
- Configuring the Redirect Server (SRC CLI) on page 167
- Configuring the Redirect Server (C-Web Interface)
- Redirect Server Redundancy on page 163
- Overview of Traffic Redirection on page 161

Configuring the Redirect Server (SRC CLI)

The redirect server on a C-series Controller manages IP layer redirection.

To configure the redirect server:

1. Configure general properties for the redirect server.

See “Configuring General Properties for the Redirect Server (SRC CLI)” on page 168 .

2. Configure a connection from the redirect server to the directory.

See “Configuring a Connection Between the Redirect Server and the Directory (SRC CLI)” on page 169 .

3. (Optional) Define traffic to be forwarded to the redirect server. In most cases you can accept the default values—traffic destined for port 80 (Web requests) and forwarded from all interface on a C-series Controller.

See “Defining Traffic to Transmit to the Redirect Server (SRC CLI)” on page 170 .

4. (Optional) Configure the number of requests that the redirect server accepts.

See “Changing the Number of Requests That the Redirect Server Accepts (SRC CLI)” on page 171 .

5. (Optional) Configure the types of files for which the redirect server accepts requests.

See “Specifying Extensions for Files That the Redirect Server Accepts (SRC CLI)” on page 172 .

6. (Optional) For a configuration to support HTTP proxies, configure DNS. You can configure the DNS server included with the redirect server, or another DNS server

on your network. If you use another DNS server, you do not need to configure the DNS server included with the redirect server.

For information about configuring the DNS server included with the redirect server, see “Configuring the DNS Server for the Redirect Server (SRC CLI)” on page 174.

7. (Optional) Configure support for HTTP proxies.

See “Verifying Configuration for the Redirect Server (SRC CLI)” on page 173.

8. (Optional) Configure a redundant redirect server.

See “Configuring a Redundant Redirect Server (SRC CLI)” on page 176.

9. Enable the redirect server.

See “Enabling the Redirect Server” on page 173.

- Related Topics**
- Configuration Statements for the Redirect Server (SRC CLI) on page 165
 - Configuring the Redirect Server (C-Web Interface)
 - Viewing Statistics for the Redirect Server (SRC CLI)
 - Overview of Traffic Redirection on page 161
 - Redirect Server Redundancy on page 163

Configuring General Properties for the Redirect Server (SRC CLI)

Use the following configuration statements to configure general properties for the redirect server:

```
redirect-server {
  destination-url destination-url;
  tcp-port tcp-port;
  refresh;
}
```

To configure properties for the redirect server:

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

```
user@host# edit redirect-server
```

2. Specify the URL to which to send subscriber traffic.

```
[edit redirect-server]
user@host# set destination-url destination-url
```

3. (Optional) Specify the TCP port on which the redirect server listens for requests.

```
[edit redirect-server]
user@host# set tcp-port tcp-port
```


4. (Optional) Specify whether the redirect server sends an HTTP 200 OK response with an HTML document that includes the `< HTTP-Equiv = "Refresh" >` header to a subscriber's browser in response to a captured request.

```
[edit redirect-server]
user@host# set refresh
```

If you do not use the **refresh** option, the redirect server sends an HTTP 302 Found response to a subscriber's browser in response to a captured request.

By setting the refresh option, the load on the Web server is decreased because non-browser (or non-HTML) client applications that use HTTP do not follow this refresh message; however, most client applications do follow HTTP 302 messages.

- Related Topics**
- Configuring the Redirect Server (SRC CLI) on page 167
 - Configuring General Properties for the Redirect Server (C-Web Interface)
 - Verifying Configuration for the Redirect Server (SRC CLI) on page 173
 - Overview of Traffic Redirection on page 161

Configuring a Connection Between the Redirect Server and the Directory (SRC CLI)

Use the following configuration statements to configure a connection between the redirect server and the directory:

```
redirect-server ldap {
  url url;
  bind-dn bind-dn;
  bind-password bind-password;
  base-dn base-dn;
}
```

To configure a connection between the redirect server and the directory:

1. From configuration mode, access the configuration statement that configures the connection.

```
user@host# edit redirect-server ldap
```

2. List the URLs for directories employed by the redirect server.

```
[edit redirect-server ldap]
user@host# set url url
```

For each URL, use the format:

```
ldap:// <host> : <portNumber>
```

where `<host>` is the IP address or hostname of the directory host and `<portNumber>` is the TCP port

3. Specify the DN that the redirect server uses to authorize connections to the directory.

```
[edit redirect-server ldap]
user@host# set bind-dn bind-dn
```

The DN must have authorization to read from *o = network*, *o = umc* in the directory.

4. Specify the password that the redirect server uses to bind to the directory.

```
[edit redirect-server ldap]
user@host# set bind-password bind-password
```

5. Specify the base DN that is the root of the directory tree.

```
[edit redirect-server ldap]
user@host# set base-dn base-dn
```

- Related Topics**
- Configuring the Redirect Server (SRC CLI) on page 167
 - Configuring a Connection Between the Redirect Server and the Directory with the C-Web Interface
 - Verifying Configuration for the Redirect Server (SRC CLI) on page 173
 - Overview of Traffic Redirection on page 161

Defining Traffic to Transmit to the Redirect Server (SRC CLI)

You can define traffic to be forwarded to the redirect server by identifying the destination port number (typically, port 80 for Web requests) for packets and the physical interface on a C-series Controller from which subscriber traffic is forwarded to the redirect server. In most cases you can accept the default values for configuration for IP redirection. If you do not specify an interface, traffic is accepted on all interfaces.

Use the following configuration statements to define traffic to transmit to the redirect server:

```
redirect-server ip-redirect{
  interface interface;
  port port;
}
```

To change the values of the port for traffic and/or the C-series interface on which traffic is forwarded to the redirect server:

1. From configuration mode, access the configuration statement that configures IP redirection for the redirect server.

```
user@host# edit redirect-server ip-redirect
```

2. Specify one or more interfaces on which subscriber traffic is forwarded from the B-RAS to the C-series Controller.

```
[edit redirect-server ip-redirect]
user@host# set interface interface
```

If you do not specify an interface, the C-series Controller accepts traffic from all interfaces.

3. Specify the TCP port of the redirected traffic. If you do not specify a port, the redirect server uses port 80 (HTTP).

```
[edit redirect-server ip-redirect]
user@host# set port port
```

- Related Topics**
- Configuring the Redirect Server (SRC CLI) on page 167
 - Defining Traffic to Transmit to the Redirect Server (C-Web Interface)
 - Verifying Configuration for the Redirect Server (SRC CLI) on page 173
 - Overview of Traffic Redirection on page 161

Changing the Number of Requests That the Redirect Server Accepts (SRC CLI)

If you want to change the number of redirection requests that the redirect server accepts, change the values for the request rates and the client rates.

Use the following configuration statements to configure the number of requests that the redirect server accepts:

```
redirect-server {
  request-rate request-rate;
  request-burst-size request-burst-size;
  client-rate client-rate;
  client-burst-size client-burst-size;
}
```

To configure the number of redirection requests that the redirect server can accept:

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

```
user@host# edit redirect-server
```

2. Specify the number of requests that the redirect server can accept per minute from all clients (global sustained rate).

```
[edit redirect-server]
user@host# set request-rate request-rate
```

3. Specify the maximum number of requests that the redirect server can accept from all clients (burst size).

```
[edit redirect-server]
user@host# set request-burst-size request-burst-size
```

This value should exceed the value for the request rate. If the value for the request rate exceeds this value, the redirect server drops the excess requests.

4. Specify the number of requests that the redirect server can accept per minute for a single client (per-client sustained rate).

```
[edit redirect-server]
user@host# set client-rate client-rate
```

5. Specify the maximum number of requests that the redirect server can accept for a single client (per client burst size).

```
[edit redirect-server]
user@host# set client-burst-size client-burst-size
```

This value should exceed the value for the client rate.

- Related Topics**
- Configuring the Redirect Server (SRC CLI) on page 167
 - Changing the Number of Requests That the Redirect Server Accepts (C-Web Interface)
 - Verifying Configuration for the Redirect Server (SRC CLI) on page 173
 - Overview of Traffic Redirection on page 161

Specifying Extensions for Files That the Redirect Server Accepts (SRC CLI)

If you do not specify the types of files that the redirect server accepts, the redirect server accepts all file types. You can identify file types by specifying the file extensions for the files that the redirect server is to accept.

Use the following configuration statements to configure the file extensions that the redirect server accepts:

```
redirect-server {
  check-file-extensions;
  file-extensions file-extensions;
}
```

To specify the extensions for the types of files accepted by the redirect server:

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

```
user@host# edit redirect-server
```

2. Specify whether the redirect server should accept only URLs that point to files that have standard file extensions— <empty>, .asp, .htm, .html, .jsp, .php, .shtm, .shtml, and .xml.

```
[edit redirect-server]
user@host# set check-file-extensions
```

If you enable check-file-extensions and the file does not have a standard file extension, the redirect server returns an HTTP 403 Forbidden message.

3. List file extensions to augment the standard file extensions you configured . Precede each extension with a period. Make sure that you specify the correct case for each character; entries are case-sensitive.

```
[edit redirect-server]
user@host# set file-extensions file-extensions
```

Separate each file extensions by a comma. For example:

```
set file-extensions .cgi,.aspx
```

- Related Topics**
- Configuring the Redirect Server (SRC CLI) on page 167
 - Specifying Extensions for Files That the Redirect Server Accepts (C-Web Interface)
 - Verifying Configuration for the Redirect Server (SRC CLI) on page 173
 - Overview of Traffic Redirection on page 161

Verifying Configuration for the Redirect Server (SRC CLI)

Purpose Verify the configuration for the redirect server.

Action At the [edit redirect-server] hierarchy level, enter the **show** command:

```
[edit redirect-server]
user@host# show
tcp-port 8800;
destination-url ;
refresh;
refresh-document etc/refresh.html;
user-name nobody;
request-rate 12000;
request-burst-size 18000;
client-rate 25;
client-burst-size 50;
```

- Related Topics**
- Configuring the Redirect Server (SRC CLI) on page 167
 - Viewing Statistics for the Redirect Server (SRC CLI)
 - Viewing Statistics for Filtered Traffic
 - Overview of Traffic Redirection on page 161

Enabling the Redirect Server

To enable the redirect server:

```
user@host> enable component redir
```

- Related Topics**
- Before You Configure the Redirect Server on a C-Series Controller on page 166
 - Configuring the Redirect Server (SRC CLI) on page 167
 - Overview of Traffic Redirection on page 161

Configuring the DNS Server for the Redirect Server (SRC CLI)

A DNS server is required to support HTTP proxies to resolve the name of any HTTP proxy, even if the name is valid only in the private domain of the client. You can use an external DNS or the DNS server that is included with the redirect server for this purpose.

If you plan to use an external DNS server, you can skip this section. This section describes how to configure the DNS server that is included with the redirect server.

Use the following configuration statements to configure the DNS server that is included with the redirect server:

```
redirect-server dns {
    enable;
    tcp-port tcp-port;
    udp-port udp-port;
    forwarder forwarder;
    error-ip-address error-ip-address;
}
```

To configure DNS for the redirect server that is included with the redirect server:

1. From configuration mode, access the configuration statement that configures DNS for the redirect server.

```
user@host# edit redirect-server dns
```

2. Enable DNS for the redirect server.

```
[edit redirect-server dns]
user@host# set enable
```

3. Specify the TCP port on which the DNS server listens:

If you set the value to 0, no TCP socket is opened.

```
[edit redirect-server dns]
user@host# set tcp-port tcp-port
```

4. Specify the UDP port on which the DNS server listens.

```
[edit redirect-server dns]
user@host# set udp-port udp-port
```

5. Specify the IP addresses of DNS servers to which resolution requests are forwarded; use commas to separate addresses, but do not add a space after the comma.

```
[edit redirect-server dns]
user@host# set forwarder forwarder
```

For example:

```
[edit redirect-server dns]
user@host# set forwarder 192.0.2.24,192.0.4.25
```

If you do not specify DNS servers, DNS resolves incoming requests by using the normal DNS method.

6. Specify the IP address that is returned when a DNS request results in an unknown name (NXDOMAIN) error.

```
[edit redirect-server dns]
user@host# set error-ip-address error-ip-address
```

- Related Topics**
- Configuring the DNS Server for the Redirect Server (C-Web Interface)
 - Before You Configure the Redirect Server on a C-Series Controller on page 166
 - Configuring the Redirect Server (SRC CLI) on page 167
 - Overview of Traffic Redirection on page 161

Configuring the Redirect Server to Support HTTP Proxies (SRC CLI)

Support for proxy requests is an optional feature of the redirect server. If you configure proxy support, you must also have DNS configured. You can use DNS servers already installed on your network, or use the server included with the SRC software.

Use the following configuration statements to configure the redirect server to support HTTP proxies:

```
redirect-server {
  proxy-support;
  proxy-destination-url proxy-destination-url;
}
```

To configure the redirect server to support HTTP proxies:

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

```
user@host# edit redirect-server
```

2. Enable HTTP proxy support.

```
[edit redirect-server]
user@host# set proxy-support
```

3. Specify the URL sent as a response to proxy requests.

```
[edit redirect-server]
```

```
user@host# set proxy-destination-url proxy-destination-url
```

If you do not configure a value, then the URL defaults to the `redir.url` value. You can use this property to send proxy requests to a page different from the direct request page on the captive portal.

- Related Topics**
- Before You Configure the Redirect Server on a C-Series Controller on page 166
 - Configuring the Redirect Server (SRC CLI) on page 167
 - Configuring the Redirect Server to Support HTTP Proxies (C-Web Interface)
 - For information about configuring the DNS server included with the SRC software, see Configuring the DNS Server for the Redirect Server (SRC CLI) on page 174
 - Overview of Traffic Redirection on page 161

Before You Configure Redundancy for a Redirect Server

If you plan to use a redundant configuration for the redirect server, ensure that:

- If you use a next-hop address for policies that capture web traffic and send it to the redirect server, that the virtual IP address to be used is also the next-hop address.
- The redirect server has SNMP write access to the virtual routers connected to it. Each VR must have at least a write community configured. (The static route from the virtual IP address to the server's real IP address is installed on the router through SNMP.)
- If additional access controls are enabled on the JUNOSe router, the hosts on which the redirect server runs must be included.

- Related Topics**
- Configuring a Redundant Redirect Server (SRC CLI) on page 176
 - Configuring a Redundant Redirect Server (C-Web Interface)
 - Overview of Traffic Redirection on page 161
 - Redirect Server Redundancy on page 163

Configuring a Redundant Redirect Server (SRC CLI)

Although configuration of a redundant redirect server is optional, we recommend that you configure redundancy to maintain high availability for the server.

Before you configure the redirect server, review configuration prerequisites. See “Before You Configure Redundancy for a Redirect Server” on page 176.

Use the following configuration statements to configure redundancy for the redirect server:


```

redirect-server {
    redundancy;
}
redirect-server monitor {
    redundant-host-ip-address redundant-host-ip-address;
    virtual-ip-address virtual-ip-address;
    real-ip-address real-ip-address;
    primary-server;
    check-interval check-interval;
    virtual-routers virtual-routers;
}

```

To configure redundancy for the redirect server:

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

```
user@host# edit redirect-server
```

2. Enable redundancy for the redirect server.

```
[edit redirect-server]
user@host# set redundancy
```

3. Configure redundancy properties for the redirect server.

```
[edit redirect-server]
user@host# edit redirect-server monitor
```

4. Configure the IP address or hostname of the redundant redirect server.

```
[edit redirect-server]
user@host# set redundant-host-ip-address redundant-host-ip-address
```

5. Configure the virtual IP address of the redirect server.

```
[edit redirect-server]
user@host# set virtual-ip-address virtual-ip-address
```

6. Configure the real IP address of the redirect server.

```
[edit redirect-server]
user@host# set real-ip-address real-ip-address
```

When a primary redirect server is started, it dynamically establishes and maintains a static route on the client router to which it connects. The static route directs traffic destined for the virtual IP address of the server to the real IP address of the active redirect server.

7. (Optional) Set the system on which you enter the command as the primary redirect server.

```
[edit redirect-server]
user@host# set primary-server
```

8. (Optional) Set the interval at which the redirect server polls the redundant redirect server.

```
[edit redirect-server]
user@host# set check-interval check-interval
```

A shorter time in the range leads to faster detection of problems and results in higher consumption of CPU resources.

9. List of virtual routers to which the redirect server connects.

```
[edit redirect-server]
user@host# set virtual-routers vrName@routerName, vrName@routerName ...
```

- Related Topics**
- Configuring the Redirect Server (SRC CLI) on page 167
 - Configuring the Virtual IP Address (SRC CLI)
 - Configuring a Redundant Redirect Server (C-Web Interface)
 - Overview of Traffic Redirection on page 161
 - Redirect Server Redundancy on page 163

Configuring Logging for the Redirect Server

The redirect server logs incoming HTTP requests through syslog with a priority of INFO and log facility of LOCAL7.

- Related Topics**
- Configuring a Component to Store Log Messages in a File (SRC CLI)
 - Configuring System Logging (SRC CLI)

Changing the Configuration for the Redirect Server

When you change the configuration for the redirect server and commit that configuration, the redirect server is automatically restarted.

- Related Topics**
- Configuring the Redirect Server (SRC CLI) on page 167
 - Overview of Traffic Redirection on page 161

Assessing Load for Redirect Server (C-Web Interface)

Purpose View the number of requests sent to the redirect server, and whether the requests reach the configured limit for the server and for server users. You can then use this information to fine-tune the properties for redirect server.

Action 1. Click **Monitor > Redirect Server > Statistics**.

The Redirect Server Statistics pane appears.

2. From the Output Style list, select an output style as described in the Help text in the main pane.
3. Click **OK**. The Redirect Server pane displays the following statistics:
 - Uptime
 - Accepted requests
 - Rejected requests
 - Number of user-limit leaky buckets
 - Number of user limits reached
 - Number of global limits reached

You can also obtain statistics for redirect server through SNMP. The name of the MIB for redirect server is Juniper-SDX-REDIRECTOR-MIB.

- Related Topics**
- Configuring General Properties for the Redirect Server (SRC CLI) on page 168
 - Viewing Statistics for the Redirect Server (C-Web Interface)
 - Viewing Information About Filtered Traffic (C-Web Interface)
 - Overview of Traffic Redirection on page 161

Part 3

Integrating JUNOS VPNs in to an SRC Configuration

- Adding VPNs from JUNOS Routing Platforms (SRC CLI) on page 183

Chapter 11

Adding VPNs from JUNOS Routing Platforms (SRC CLI)

- Before You Add a JUNOS VPN to the SRC Configuration on page 183
- Configuring VPNs to Integrate into an SRC Network on page 184
- Configuration Statements for Adding VPNs and Extranet Clients on page 184
- Adding VPNs for Retailers and Enterprises on page 185
- Verifying and Updating Configuration of Extranets for VPNs on page 186
- Locating and Removing Inactive Subscriptions to a VPN on page 187

Before You Add a JUNOS VPN to the SRC Configuration

Before you can add a VPN to an SRC configuration, you must configure the VPN. Before you configure the VPN, make sure that in the routing scheme in the VPN:

- All members in the VPN can reach other.
- No changes are needed as members are added to and removed from the VPN.

If a VPN is used as an intranet, you can ensure that the routing scheme meets these requirements by configuring either:

- Static routes in the VPN
- Appropriate routing protocols

If the VPN is exported as an extranet, some members of the VPN may use private or conflicting address schemes. In addition, if the VPN has a large number of potential members, configuring static routing or routing protocols for all potential members may not be a manageable proposition. In these last two cases, we recommend that you use public addresses in the VPN and have VPN members implement Network Address translation (NAT) for traffic destined for the VPN.

VPNs use private IP addresses. If, however, enterprises that you administer export VPNs to extranet clients, you must ensure that the extranet clients can reach the IP addresses that the VPNs use. To implement an address scheme that allows all subscribers who have access to a VPN, we recommend that you implement NAT on the JUNOS routing platform. IT managers in the retailers and enterprises who own the VPNs can then map private IP addresses in the VPNs to public IP addresses, which extranet clients can reach.

Before you can reference a JUNOS VPN from the SRC configuration:

1. Create one routing instance in each router where VPN members access the VPN.
2. Make sure that each routing instance in the VPN has the same name as the VPN. The VPN represents the collection of the routing instances, the VPN members, and the connections between those routing instances within the VPN. All routing instances share a VPN ID, which you use to add VPNs to an SRC configuration.
3. Connect the VPN through a tunnel such as an MPLS label-switched path or IP Security tunnel.

- Related Topics**
- Overview of NAT Address Management Portal
 - Assigning IP Addresses

Configuring VPNs to Integrate into an SRC Network

For SRC configurations that support JUNOS routers, you can add VPNs and extranets for retailers and enterprises.

For C-series Controllers, you add VPNs through the CLI and can manage the VPNs through an enterprise portal that runs on another system.

- Related Topics**
- Adding VPNs for Retailers and Enterprises on page 185

Configuration Statements for Adding VPNs and Extranet Clients

Use the following configuration statements to add VPNs and extranet clients at the [edit] hierarchy level.

```
subscribers retailer name vpn vpn-id {
    description description ;
    display-name display-name ;
    extranet-client [ extranet-client ... ];
    imported-extranet [ imported-extrane t...];
}
subscribers retailer name subscriber-folder folder-name enterprise name vpn vpn-id
{
    description description ;
    display-name display-name ;
    extranet-client [ extranet-client ... ];
    imported-extranet [ imported-extrane t...];
}
```

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

Adding VPNs for Retailers and Enterprises

When you add a VPN to the SRC configuration, you are creating a VPN configuration object that represents a VPN that is already configured in the network. You can add a VPN for a retailer or for an enterprise.

Before you add a VPN to the configuration, obtain the identifier for the VPN. This identifier is the name of the routing instances on a JUNOS routing platform that implements the VPN.

To add a VPN to subscriber configuration for a retailer or an enterprise:

1. From configuration mode, access the configuration statement that configures the VPN.

```
[edit]
user@host# edit subscribers retailer name vpn vpn-id
```

or

```
[edit]
user@host# edit subscribers retailer name subscriber-folder folder-name
enterprise name vpn vpn-id
```

where *vpn-id* is the name of the routing instances on a JUNOS routing platform that implements the VPN.

2. (Optional) Provide a name to identify the VPN as it appears in other SRC components, such as the Enterprise Manager Portal or other login pages.

```
[edit subscribers retailer name vpn vpn-id ]
user@host# edit display-name display-name
```

For example, to label the VPN as one used for video conferences with corporate partners:

```
[edit subscribers retailer name vpn vpn-id ]
user@host# edit display-name " Partner Video Conference"
```

3. (Optional) Add a description of the VPN.

```
[edit subscribers retailer name vpn vpn-id ]
user@host# edit description description
```

For example:

```
[edit subscribers retailer name vpn vpn-id ]
user@host# edit description " VPN for video conference with partners"
```

4. Verify that the configuration is correct. For example:

```
[edit subscribers retailer Acme vpn 1234]
user@host# show
```

```
display-name "Partner Video Conference";
description "VPN for video conference with partners.";
```

Verifying and Updating Configuration of Extranets for VPNs

From the SRC CLI, you can correct errors in extranet configuration when these errors result from directory or portal errors. In the extranet configuration, an extranet client of an object must be imported by that object.

In the SRC configuration for a subscriber that is the client of an extranet client, you specify a VPN for the imported extranet client. Typically, you add the extranet client and specify the imported extranet from the Enterprise Manager Portal. You can use the SRC CLI to verify the configuration and to make updates to the existing configuration.

To view information about extranet configuration and update it:

1. From configuration mode, access the configuration statement that represents the configuration for the VPN.

```
[edit]
user@host# edit subscribers retailer name vpn vpn-id
```

or

```
[edit]
user@host# edit subscribers retailer name subscriber-folder folder-name
enterprise name vpn vpn-id
```

where *vpn-id* is the name of the routing instances on a JUNOS routing platform that implements the VPN.

2. View the configuration for the VPN. For example:

```
[edit subscribers retailer Acme vpn 1234]
user@host# show
extranet-client [ "enterpriseName=Acme, ou=local, retailername=default,
o=Users,
o=umc" "enterpriseName=WidgetCo, ou=local, retailername=default, o=Users,
o=UMC "];
```

3. (Optional) Change or add the distinguished name (DN) of a retailer or an enterprise that is an extranet client of this VPN.

```
[edit subscribers retailer name vpn vpn-id ]
user@host# set extranet-client extranet-client
```

For example:

```
[edit subscribers retailer name vpn vpn-id ]
user@host# set extranet-client
enterpriseName=Acme2,ou=local,retailername=default, o=Users, o=umc
```

4. (Optional) Change or add extranets to be imported by specifying the DN of the extranet.

```
[edit subscribers retailer name vpn vpn-id ]
user@host# set imported-extranets imported-extranets
```

You can specify one or more extranets.

5. Verify that the updated configuration is correct.

```
[edit subscribers retailer name vpn
vpn-id
]
user@host# show
[edit subscribers retailer Acme vpn 1234]
user@host# show
extranet-client [ "enterpriseName=Acme, ou=local, retailername=default,
o=Users,
o=umc" "enterpriseName=Acme2, ou=local, retailername=default, o=Users,
o=umc""enterpriseName=WidgetCo, ou=local, retailername=default, o=Users,
o=UMC "];
```

Locating and Removing Inactive Subscriptions to a VPN

When an IT manager cancels the export of a VPN, the Enterprise Manager Portal automatically deactivates any active subscriptions to that VPN for the associated extranet client. If an IT manager cancels the export of a VPN at the same time that the extranet client activates a subscription to this VPN, there is a remote possibility that the Enterprise Manager portal will maintain the active subscription.

We recommend that you periodically check for and deactivate these types of invalid subscriptions to prevent this type of invalid subscription.

Part 4

Index

- Index on page 191

Index

A

access lines.....	6
description.....	4, 5
accesses	
configuring subscriptions	
SRC CLI.....	155
accounting	
basic RADIUS accounting plug-in.....	95
custom RADIUS accounting plug-ins.....	95
flat file accounting plug-ins.....	95
flexible RADIUS accounting plug-ins.....	96
anonymous subscriber.....	19
authenticated subscriber.....	19
authentication plug-ins	
configuring	
SRC CLI.....	106
types.....	82
authorization plug-ins	
configuring	
SRC CLI.....	106
types.....	82

B

basic RADIUS accounting plug-in.....	95
configuring	
SRC CLI.....	99
basic RADIUS authentication plug-in.....	107
configuring	
SRC CLI.....	108

C

captive portal	
preventing access to resources.....	161
classification scripts	
conditions.....	43
glob matching.....	46
joining.....	46
regular expression matching.....	48
configuring	
SRC CLI.....	46
descriptions.....	43

DHCP classification, C-series controller	
configuring, SRC CLI.....	68
description.....	43
targets.....	72
DHCP classification, C-series Controller	
conditions.....	70
interface classification, C-series controller	
conditions.....	52
configuring, SRC CLI.....	49
description.....	43
examples.....	52
how it works.....	43
targets.....	52
structure	
SRC CLI.....	46
subscriber classification, C-series controller	
condition.....	59
configuring, SRC CLI.....	56
description.....	43
DHCP options.....	63
enterprise subscriber example.....	65
how it works.....	43
static IP subscriber example.....	65
subscriber group example.....	65
targets.....	64
target, C-series controller	
definition.....	43
expressions.....	46
types.....	46
component interactions	
DHCP	
initial login.....	18
persistent login.....	21
subscriber account login.....	19
subscriber logout.....	22
enterprise subscribers	
login.....	26
remote session activation.....	33
PPP	
login.....	13
logout.....	15
static IP subscribers.....	23
subscription activation.....	29
subscription deactivation.....	31

conventions	
notice icons.....	xix
text.....	xix
COPS (Common Open Policy Service)	
DHCP interactions	
initial login.....	18
logout.....	22
persistent login	21
subscriber account login	20
interface startup interactions.....	17
PPP interactions	
login.....	14
logout.....	16
static IP subscriber interactions.....	23
subscription activation interactions.....	30
subscription deactivation interactions.....	31
custom RADIUS accounting plug-ins.....	95
configuring	
SRC CLI.....	104
custom RADIUS authentication plug-ins.....	107
configuring	
SRC CLI.....	112
customer support.....	xxiii
contacting JTAC.....	xxiii

D

default retailer authentication plug-ins	
configuring	
SRC CLI.....	131
default retailer DHCP authentication plug-ins	
configuring	
SRC CLI.....	131
denial-of-service attacks.....	163
DHCP (Dynamic Host Configuration Protocol)	
address assignment.....	84
classification scripts. <i>See</i> classification scripts	
options.....	74
profiles	
SRC CLI.....	77
subscribers	
login process.....	17
logout process.....	22
documentation set	
comments on.....	xxiii

E

enterprise	
description.....	4
enterprise subscribers.....	3
adding	
SRC CLI.....	145
enterprise subscribers, login process	26

event publishers	
configuring	
SRC CLI.....	130
default retailer authentication, configuring	
SRC CLI.....	131
default retailer DHCP authentication, configuring	
SRC CLI.....	131
description.....	81
retailer-specific.....	130
service-specific.....	130
virtual router-specific.....	131
example-simple.....	54, 55, 65, 66, 67, 68
external plug-ins	
configuring	
SRC CLI.....	90

F

flat file accounting plug-ins.....	95
configuring	
SRC CLI.....	96
configuring headers	
SRC CLI.....	98
flexible RADIUS accounting plug-ins.....	96
attributes, defining	
SRC CLI.....	118
configuring.....	101
RADIUS packets, defining.....	117
flexible RADIUS authentication plug-ins.....	107
attributes, defining	
examples.....	127
SRC CLI.....	118
configuring	
SRC CLI.....	110
RADIUS packets, defining	
SRC CLI.....	117
setting responses	
SRC CLI.....	127

G

general properties	
configuring	
SRC CLI.....	168

H

HTTP proxy.....	161
-----------------	-----

I

interface classification scripts. <i>See</i> classification scripts	
interim accounting, configuring on SAE.....	37
internal plug-ins	
configuring	
SRC CLI.....	89

L

LDAP authentication plug-in.....	107
configuring	
SRC CLI.....	115
limiting subscribers plug-in.....	107
configuring	
SRC CLI.....	108
logging	
redirect server.....	178
login events, description.....	10
login process	
enterprise.....	26
residential.....	9, 11
DHCP.....	17
PPP.....	13
<i>See also</i> logout process, residential	
summary.....	11
login registration	
configuring	
SRC CLI.....	40
logout process, residential	
DHCP.....	22

M

managers	
configuring	
SRC CLI.....	151
control over all retailers.....	7
management privileges.....	6
subscribers and subscriptions.....	6
manuals	
comments on.....	xxiii

N

NAT (Network Address Translation)	
VPNs.....	183
notice icons.....	xix

P

plug-ins	
activating service sessions.....	87
authentication	
configuring, SRC CLI.....	106
authorization	
configuring, SRC CLI.....	106
basic RADIUS accounting.....	95
configuring, SRC CLI.....	99
basic RADIUS authentication.....	107
configuring, SRC CLI.....	108
creating subscriber sessions.....	86
custom RADIUS accounting.....	95
configuring, SRC CLI.....	104

custom RADIUS authentication.....	107
configuring, SRC CLI.....	112
defining RADIUS packets	
SRC CLI.....	117
DHCP address assignment.....	84
event publishers. <i>See</i> event publishers	
external	
configuring, SRC CLI.....	90
flat file accounting.....	95
configuring, SRC CLI.....	96
flexible RADIUS accounting.....	96
configuring.....	101
flexible RADIUS authentication.....	107
configuring, SRC CLI.....	110
internal.....	82
authorization.....	82
configuring RADIUS peers, SRC CLI.....	93
configuring, SRC CLI.....	89
customizing RADIUS packets.....	82
how they work.....	81
pool.....	81
RADIUS attributes, SRC CLI.....	118
tracking.....	82
LDAP authentication.....	107
configuring, SRC CLI.....	115
limiting subscribers.....	107
configuring, SRC CLI.....	108
state synchronization	
configuring, SRC CLI.....	91
tracking	
configuring, SRC CLI.....	95
service sessions.....	87
subscriber sessions.....	86
PPP subscribers	
login process.....	13
Web login.....	13
prevention, use of unauthorized resources.....	161
protocols	
routing.....	183
proxy HTTP.....	161
proxy request management.....	161
public addresses, VPNs.....	183

Q

QoS tracking plug-in.....	96
---------------------------	----

R

RADIUS attributes	
defining in RADIUS plug-ins	
SRC CLI.....	118
examples, defining in RADIUS plug-ins	
SRC CLI.....	127
RADIUS client library, custom RADIUS plug-ins.....	82
RADIUS packets, customizing in plug-ins.....	82

RADIUS peers	
configuring in plug-ins	
SRC CLI.....	93
RADIUS plug-ins.....	95
authentication.....	107
UDP port.....	117
<i>See also</i> plug-ins	
redirect server	
assessing load	
C-Web interface.....	178
configuration statements	
SRC CLI.....	165
configuring	
SRC CLI.....	167
configuring DNS server for	
SRC CLI.....	174
configuring HTTP proxy support	
SRC CLI.....	175
configuring redundant	
SRC CLI.....	176
directory connection	
SRC CLI.....	169
failover.....	163
file extensions	
SRC CLI.....	172
logging	
SRC CLI.....	178
number of requests	
SRC CLI.....	171
protection against denial-of-service attacks.....	163
redundancy.....	163, 176
static route to router.....	163
traffic definition	
SRC CLI.....	170
verifying	
SRC CLI.....	173
redundancy	
redirect server.....	163
residential subscribers.....	3
adding	
SRC CLI.....	141
login process. <i>See</i> login process	
retailers	
subscribers.....	3
adding, SRC CLI.....	137
router subscribers.....	4
adding	
SRC CLI.....	149
routing instances	
VPNs.....	183
routing scheme.....	183

S

SAE (service activation engine)	
classification scripts. <i>See</i> classification scripts	
login events.....	9
login process. <i>See</i> login process	
SAE (service activation engine), configuring	
interim accounting	
SRC CLI.....	37
login registration	
SRC CLI.....	40
multiple logins from same IP address	
SRC CLI.....	39
reduce reported session time	
SRC CLI.....	38
session reactivation timers	
SRC CLI.....	40
time for MAC address in cache	
SRC CLI.....	35
unauthenticated user DN	
SRC CLI.....	36
service activation engine. <i>See</i> SAE	
service sessions	
activate-on-login.....	32, 87
activating and tracking.....	87
activating with Web application.....	28
enterprise, remote activation.....	33
sites.....	5, 6
subscriber.....	3
adding, SRC CLI.....	148
state synchronization plug-in interface	
configuring	
SRC CLI.....	91
static IP subscribers, login process.....	23
static routing.....	183
subscriber classification scripts. <i>See</i> classification scripts	
subscriber folders.....	4
adding	
SRC CLI.....	140
subscriber sessions	
activating with Web application.....	29
creating and tracking.....	86
enterprise, creating and activating.....	26
subscribers	
adding	
SRC CLI.....	137
enterprise.....	3
adding, SRC CLI.....	145
inheriting properties.....	136
inheriting subscriptions.....	136
residential.....	3
adding, SRC CLI.....	141
retailer.....	3
adding, SRC CLI.....	137
router.....	4
adding, SRC CLI.....	149

sites.....	3
adding, SRC CLI.....	148
types.....	3
subscriptions.....	4
access, configuring	
SRC CLI.....	155
activation order, specifying	
SRC CLI.....	135
multiple per subscriber.....	153
support, technical <i>See</i> technical support	

T

targets. <i>See</i> classification scripts	
technical support	
contacting JTAC.....	xxiii
text conventions defined.....	xix
tracking plug-ins.....	82
configuring	
SRC CLI.....	95

U

UDP ports	
RADIUS plug-ins.....	117
User Datagram Protocol. <i>See</i> UDP	

V

validating	
VPNs.....	187
virtual private networks. <i>See</i> VPNs	
VPNs (virtual private networks)	
adding	
SRC CLI.....	185
configuration requirements.....	183
configuration statements.....	184
definition.....	184
extranet clients, modifying	
SRC CLI.....	186
invalid subscriptions.....	187
modifying.....	186
routing schemes.....	183
using NAT.....	183
validating.....	187

