SRC-PE Software

Integration Guide: Network Devices, Directories, and RADIUS Servers

Release 1.0.x

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

Part Number: 530-020342-01, Revision A00

SRC-PE Software Integration Guide: Network Devices, Directories, and RADIUS Servers, Release 1.0.x
Writing: Linda Creed, Justine Kangas, Betty Lew, Helen Shaw, Brian Wesley Simmons, Michael Taillon
Editing: Fran Mues
Illustration: Nathaniel Woodward
Cover Design: Edmonds Design

Revision History
6 April 2007—Revision 1

Software License

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

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

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

END USER LICENSE AGREEMENT

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

1. The Parties. The parties to this Agreement are Juniper Networks, Inc. and its subsidiaries (collectively "Juniper"), and the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. The Software. In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, and updates and releases of such software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller. "Embedded Software" means Software which Juniper has embedded in the Juniper equipment.

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

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

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

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface,

processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Table of Contents

About This Guide xi

Objectives ... xi
Audience ... xi
Documentation Conventions ... xii
Related Juniper Networks Documentation ... xiii
Obtaining Documentation ... xv
Documentation Feedback ... xv
Requesting Support ... xvi

Part 1 Network Devices

Chapter 1 Integrating Third-Party Network Devices into the SRC Network with the SRC CLI 3

Overview of Integrating Network Devices into the SRC Network ... 3
SAE Communities ... 4
Storing Session Data ... 4
Using Script Services to Provision Third-Party Devices ... 5
Logging In Subscribers and Creating Sessions ... 5
Assigned IP Subscribers ... 6
Login Interactions with Assigned IP Subscribers ... 6
Event Notification from an IP Address Manager ... 7
Login with Event Notification ... 7
Configuration Tasks for Integrating Third-Party Network Devices ... 9
Setting Up Script Services ... 10
Adding Objects for Network Devices ... 10
Adding Virtual Router Objects ... 11
Setting Up SAE Communities ... 12
Configuring the SAE Community Manager ... 12
Specifying the Community Manager in the SAE Device Driver ... 14
Configuring SAE Properties for the Event Notification API ... 14
Developing Initialization Scripts for Network Devices ... 15
Interface Object Fields ... 15
Required Methods ... 16
Example: Initialization Script ... 16
Copying Initialization Scripts to the C-series Platform ... 17
Specifying Initialization Scripts on the SAE ... 17
Using SNMP to Retrieve Information from Network Devices ... 18
Configuring Global SNMP Communities in the SRC Software ... 18
Using the NIC Resolver ... 18

Chapter 2 Integrating Third-Party Network Devices into the SRC Network on a Solaris Platform 19

Overview of Integrating Network Devices into the SRC Network....19
SAE Communities....20
Storing Session Data....20
Using Script Services to Provision Third-Party Devices....21
Logging In Subscribers and Creating Sessions....21
Assigned IP Subscribers....22
Login Interactions with Assigned IP Subscribers....22
Event Notification from an IP Address Manager....23
Login with Event Notification....23
Configuration Tasks for Integrating Third-Party Network Devices....25
Setting Up Script Services....26
Adding Objects for Network Devices to the Directory....26
Router Fields....27
Adding Virtual Router Objects....28
Virtual Router Fields....29
Setting Up SAE Communities....32
Defining SAE Communities....32
Adding an SAE....33
Modifying an SAE Address....33
Deleting an SAE Address....33
Configuring the SAE Community Manager....34
Community Manager Fields....34
Specifying the SAE Community Manager....35
Proxy Router Driver Fields....36
Configuring SAE Properties for the Event Notification API....36
Event API Fields....37
Developing Initialization Scripts for Network Devices....37
Interface Object Fields....37
Required Methods....39
Example: Router Initialization Script....39
Specifying Router Initialization Scripts on the SAE....39
Router Script Fields....40
Using SNMP to Retrieve Information from Network Devices....41
Configuring Global SNMP Communities in the SRC Software....41
Global SNMP Community Fields....41
Using the NIC Resolver....42

Part 2 Integrating Directories

Chapter 3 Overview of LDAP Integration 45

LDAP Overview....45
Directory Availability....46
Directory Updates....46
Supported Directories....47
Directory Security....47
Directory Access....47
LDAPS Directory Connections....48
Provisioning the Directory....48

Naming Directory Entries....48
Directory Object Model and Schema....49
Directory Object Model....49
Naming Convention for Entries....50
Directory Schema....51
Object Classes....51
Attributes....55
Structure Rules....55
Content Rules....55
Where to Find More Information About the Object Model and Directory Schema....56

Chapter 4 Integrating eTrust Directory 57

Overview of Integration with eTrust Directory....57
About the eTrust Directory Add-On Package....57
Integrating the eTrust Directory with the SRC Software....59
Installing eTrust Directory to Integrate with the SRC Software....59
Configuring the SDX eTrust Directory Server Agent with the SRC Software....59
Starting SDX eTrust Directory....60
Stopping SDX eTrust Directory....61
Displaying the Status of SDX eTrust Directory....61
Backing Up and Restoring eTrust Directory....61

Chapter 5 Integrating Oracle Internet Directory 63

Overview of Oracle Internet Directory Integration....63
About the Oracle Internet Directory Add-On Package....64
Integrating the Oracle Internet Directory with the SRC Software....64
Installing Oracle Internet Directory to Integrate with the SRC Software....65
Before You Install Oracle Internet Directory....65
Specifying Configuration Values During Installation....65
Verifying Directory Settings....66
Running the Load Script for the Oracle Internet Directory Integration....66
Starting and Stopping Oracle Internet Directory....66
Setting Up Local Configuration for SRC Components....66
Backing Up and Restoring the Oracle Internet Directory....67

Chapter 6 Integrating Sun ONE Directory Server 69

Overview of Sun ONE Directory Server Integration....69
About the Sun ONE Add-On Package....70
Silent Installation for Sun ONE Directory Server....70
Load Script to Integrate Sun ONE Directory Server....71
Integrating the Sun ONE Directory with the SRC Software....71
Installing the Sun ONE Directory Add-On Package....72
Configuring an Instance of Sun ONE Directory Server....72
Starting Sun ONE Directory Server....73
Stopping Sun ONE Directory Server....74
Restarting Sun ONE Directory Server....74
Backing Up the Sun ONE Database....74
Restoring the Sun ONE Database....75

Chapter 7 Integrating the DirX Directory Server 77

Overview of DirX Directory Server Integration ... 77
About the DirX Add-On Package ... 78
Integrating the DirX Directory with the SRC Software ... 78
Preparing to Install the DirX Directory Server ... 79
Installing the DirX Directory Server ... 80
Installing the UMCdirxa Add-On Package ... 80
Configuring the DirX Directory Server ... 80
Provisioning the Directory by Using DirXmetahub ... 81
Uninstalling the DirX Directory Server ... 82
Starting the DirX Directory Server ... 82
Starting the DirX Directory Server in a dirx user Environment ... 82
Starting the DirX Directory Server in a Superuser Environment ... 83
Stopping the DirX Directory Server ... 83
Stopping the DirX Directory Server in a dirx user Environment ... 83
Stopping the DirX Directory Server in a Superuser Environment ... 83
Backing Up the DirX Database ... 84
Restoring the DirX Directory Database ... 84

Chapter 8 Configuring LDAPS for SRC Components 85

Overview of LDAPS Support ... 85
LDAPS Authentication and Connection ... 85
Configuring LDAPS Connections ... 86
Configuring the Directory Server to Support LDAPS Connections ... 86
Establishing Trust for Directory Clients ... 87
Configuring the SAE to Find the Certificate Store ... 87
Enabling LDAPS Communication for SAE Components ... 88
Disabling LDAPS Communication for SAE Components ... 89

Chapter 9 Integrating Data with the LDAP Directory 91

Overview of Data Integration ... 91
Getting Help with Data Integration ... 93
Installing the Data Integration Suite ... 94
Planning Data Integration ... 94
Developing Data Integrators ... 94
Configuring Data Integrators ... 95
Defining Properties for the Database Reader ... 96
Defining Properties for the LDAP Reader ... 97
Defining Properties for the XML File Reader ... 98
Defining Properties for the Enterprise Audit File Reader ... 98
Defining Properties for the XML File Writer ... 99
Defining Properties for the XSLT Translator ... 99
Defining Properties for the LDAP Writer ... 99
Executing Data Integration ... 100
Examples of Data Integrators ... 100
Example: VPN Directory Updater ... 100
Example: VPN Subscription Deactivator ... 102

Chapter 10 Access Control Scheme 105

Directory Configuration ... 105
Directories ... 106
User Class ... 106

Permissions 106
Access Controls 107
Access Controls for the Entire Tree 107
Access Controls Against Objects from Type cachedAuthentication Profile and UmcConfiguration 108
Access Controls Against sspServiceProfile 108
Access Controls Against umcRadius Person and umcUser 109
Access Controls Against RADIUS Profiles 109
Access Controls Against the Policy Subtree 110
Access Controls Against the Parameter Subtree 110
Access Controls for System Management 111
Access Controls Against the Lock Subtree 111
Access Controls Against Subscriber, Retailer, and Service Profiles 112
Access Controls Against the Network Subtree 112
Access Controls Against Services and Mutex Group Objects 113
Access Controls Against the Workflow Subtree 113
Access Controls Against the User Subtree 114
Access Controls Against Service, Policy, and Global Parameter Objects 114
Activation Access Rights 115
Subscription Access Rights 115
Substitution Access Rights 116
Common Access Rights for All Managers 116
Directory-Specific Access Control Implementation 117
DirX Directory Server 117
Sun ONE Directory Server 118

Part 3 Integrating RADIUS Servers

Chapter 11 Integrating Steel-Belted Radius/SPE Server 123

System Requirements for the Steel-Belted Radius Server 124
Installing the Steel-Belted Radius/SPE Software 124
Installing the Steel-Belted Radius Software for the First Time 124
Installing the Steel-Belted Radius Software over Previous Installations 125
Enabling LDAP Authentication 126
Configuring UDP Ports for Steel-Belted Radius Software 127
Starting the Steel-Belted Radius/SPE Server 128
Stopping the Steel-Belted Radius/SPE Server 128
Extending Dictionary Files with JUNOSe Parameters for the Steel-Belted Radius Server 129
Configuring LDAP Authentication 129
[Bootstrap] Section 130
[Settings] Section 130
[Server] Section 131
[Server/serverName] Section 131
[Search/name] Section 132
[Attribute/name] Section 132
[Request] Section 134
[Response] Section 134
Configuring Directed Authentication 136
Customizing the Authentication Log File 137

Configuring the Steel-Belted Radius/SPE Server and RADIUS Clients 137
Configuring the Steel-Belted Radius Server 137
Configuring RADIUS Clients 138
Using the Radius Administrator to Configure RADIUS Clients 138

Chapter 12 Integrating Merit RADIUS 143

System Requirements for the Merit AAA Server 144
Installing Merit AAA 144
LDAP Features for the Merit AAA Server 144
Configuring UDP Ports for the Merit AAA Server 145
Starting the Merit AAA Server 146
Stopping the Merit AAA Server 147
Displaying the Status of the Merit AAA Server 147
Extending Dictionary Files with JUNOSe Parameters for the Merit AAA Server 147
Configuring LDAP Authentication for the Merit AAA Server 148
Configuring the Merit AAA Server 148
Configuring RADIUS Profiles with the LDAP Directory 151
Example: Merit AAA Accounting Log File Format 151
Configuring the Merit AAA Server and RADIUS Clients 153
Configuring the Merit AAA Server 153
Configuring RADIUS Clients 153
Testing the Merit AAA Server 154

Chapter 13 Integrating RAD-Series RADIUS Server 155

System Requirements for the RAD-Series RADIUS Server 156
Installing the RAD-Series RADIUS Server 156
LDAP Features for the RAD-Series RADIUS Server 157
Configuring UDP Ports for the RAD-Series RADIUS Server 158
Starting and Stopping RAD-Series Server Manager 159
Changing the UDP Ports 160
Extending Dictionary Files with JUNOSe Parameters for the RAD-Series RADIUS Server 161
Configuring LDAP Authentication for the RAD-Series RADIUS Server 161
Configuring the RAD-Series Server Manager 161
Configuring Realm Administration 164
Configuring LDAP Settings 165
Configuring RADIUS Profiles with the LDAP Directory 165
Example: RAD-Series RADIUS Server Accounting Log File Format 166
Configuring the RAD-Series RADIUS Server and RADIUS Clients 167
Configuring the RAD-Series RADIUS Server 167
Configuring RADIUS Clients 168
Testing the RAD-Series RADIUS Server 168

Index 169

About This Guide

This preface provides the following guidelines for using the *SRC-PE Software Integration Guide: Network Devices, Directories and RADIUS Servers*.

- Objectives on page xi
- Audience on page xi
- Documentation Conventions on page xii
- Related Juniper Networks Documentation on page xiii
- Obtaining Documentation on page xv
- Documentation Feedback on page xv
- Requesting Support on page xvi

Objectives

This guide describes how to integrate the Session and Resource Control (SRC) software with a Lightweight Directory Access Protocol (LDAP) directory server and a RADIUS AAA solution.

NOTE: If the information in the latest *SRC Release Notes* differs from the information in this guide, follow the *SRC Release Notes*.

Audience

This guide is intended for experienced system and network specialists working with JUNOSe 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. For users who deploy the SRC software on a Solaris platform, we also assume that readers are familiar with the Lightweight Directory Access Protocol (LDAP) and the UNIX operating system.

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

The sample screens used throughout this guide are representations of the screens that you will see when you install and configure the SRC software. The actual screens may differ.

For convenience and clarity, the installation and configuration examples show default file paths. If you do not accept the installation defaults, your paths will vary from the examples.

Table 1 defines notice icons used in this guide. Table 2 defines text conventions used throughout the 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.

Table 2: Text Conventions

Convention	Description	Examples
Bold typeface	■ Represents keywords, scripts, and tools in text. ■ Represents a GUI element that the user selects, clicks, checks, or clears.	■ Specify the keyword **exp-msg**. ■ Run the **install.sh** script. ■ Use the **pkgadd** tool. ■ To cancel the configuration, click **Cancel**.
Bold sans serif typeface	Represents text that the user must type.	user@host# **set cache-entry-age** *cache-entry-age*

`Monospace sans serif typeface`	Represents information as displayed on your terminal's screen, such as CLI commands in output displays.	`nic-locators {`
		`  login {`
		`  resolution {`
		`  resolver-name /realms/login/A1;`
		`  key-type LoginName;`
		`  value-type SaeId;`
		`  }`

Table 2: Text Conventions (continued)

Convention	Description	Examples
Regular sans serif typeface	■ Represents configuration statements. ■ Indicates SRC CLI commands and options in text. ■ Represents examples in procedures. ■ Represents URLs.	■ system ldap server { stand-alone; ■ Use the request sae modify device failover command with the force option. ■ user@host# . . . ■ http://www.juniper.net/techpubs/software/management/sdx/api-index.html
Italic sans serif typeface	Represents variables in SRC CLI commands.	user@host# **set local-address** *local-address*
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <gfwif>.
Key name	Indicates the name of a key on the keyboard.	Press Enter.
Key names linked with a plus sign (+).	Indicates that you must press two or more keys simultaneously.	Press Ctrl+b.
Italic typeface	■ Emphasizes words. ■ Identifies chapter, appendix, and book names. ■ Identifies distinguished names. ■ Identifies files, directories, and paths in text but not in command examples.	■ There are two levels of access: *user* and *privileged*. ■ *Chapter 2, Services.* ■ *o = Users, o = UMC* ■ The */etc/default.properties* file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	Plugin.radiusAcct-1.class = \ net.juniper.smgt.sae.plugin\ RadiusTrackingPluginEvent
Words separated by the | symbol	Represent a choice to select one keyword or variable to the left or right of this symbol. (The keyword or variable may be either optional or required.)	diagnostic | line

Related Juniper Networks Documentation

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

With each SRC Application Library release, we provide the *SRC Application Library CD*. This CD contains both the software applications and the *SRC Application Library Guide*.

The C-Web interface, which is based on the J-Web interface, is available for monitoring C-series platforms and the SRC software. For general information about the J-Web interface, see the *J-Web Interface User Guide*.

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

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

Document	Description
Core Documentation Set	
C-series Hardware Guide	Describes the hardware platforms and how to install, maintain, replace, and troubleshoot them. The guide also includes specifications.
SRC-PE Getting Started Guide	Describes the SRC software and explains how to set up an initial configuration and manage a C-series platform. The guide describes how to set up and start the SRC CLI and C-Web, as well as other SRC configurations. It provides information about setting up an initial SRC configuration on a Solaris platform. The guide also describes how to upgrade the SRC software and how to use the SRC configuration tools. It includes reference material for the SRC documentation.
SRC-PE CLI User Guide	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.
SRC-PE Network Guide: SAE, Juniper Networks Routers, and NIC	Describes how to use and configure the SAE and the NIC. This guide also provides detailed information for using JUNOSe routers and JUNOS routing platforms in the SRC network.
SRC-PE Integration Guide: Network Devices, Directories, and RADIUS Servers	Describes how to integrate external components—network devices, directories, and RADIUS servers—into the SRC network. The guide provides detailed information about integrating specific models of the external components.
SRC-PE Services and Policies Guide	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.
SRC-PE Subscribers and Subscriptions Guide	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 sample residential portals and enterprise service portals, including the Enterprise Manager Portal.
SRC-PE Monitoring and Troubleshooting Guide	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.
SRC-PE Solutions Guide	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); mirroring subscriber traffic on JUNOSe routers; demonstrating network resource management features in a sample IP television (IPTV) application; and demonstrating the integration of prepaid services in a sample application.
SRC-PE CLI Command Reference, Volume 1 *SRC-PE CLI Command Reference, Volume 2*	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.
SRC-PE Comprehensive Index	Provides a complete index of the SRC guides, excluding the *C-series Hardware Guide* and the *SRC CLI Command Reference*.
J-Web User Interface Guide	Provides general information about the J-Web interface.

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

Document	Description
Application Library	
SRC Application Library Guide	Describes how to install and work with applications that you can use to extend the capabilities of the SRC software. The guide documents the following applications: SRC-SG (SOAP Gateway) Web applications, applications to integrate the Juniper Networks Intrusion Detection and Protection (IDP) software into an SRC-managed environment, an application to provide endpoint security by integrating Juniper Networks Instant Virtual Extranet (IVE) Host Checker, a traffic-mirroring Web application, an application to integrate IP address managers with the SAE, 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, an application to control volume usage, and the SRC-ACP (Admission Control Plug-In) application.
Release Notes	
SRC-PE Release Notes *SRC Application Library Release Notes*	In the *Release Notes*, 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 *Release Notes* differs from the information found in the documentation set, follow the *Release Notes*. 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 documentation, see the products documentation page on the Juniper Networks Web site at

http://www.juniper.net/

To order printed copies of this manual and other Juniper Networks technical documents, or to order a documentation CD, which contains this manual, contact your sales representative.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation to better meet your needs. Send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at

http://www.juniper.net/techpubs/docbug/docbugreport.html

If you are using e-mail, be sure to include the following information with your comments:

- Document name
- Document part number
- Page number
- Software release version

Requesting Support

For technical support, open a support case using the Case Manager link at

http://www.juniper.net/support/

or call 1-888-314-JTAC (from the United States, Canada, or Mexico) or 1-408-745-9500 (from elsewhere).

Part 1
Network Devices

Chapter 1

Integrating Third-Party Network Devices into the SRC Network with the SRC CLI

This chapter describes how to use the SRC CLI to integrate third-party network devices into the SRC network. You can use the CLI to configure the SRC software on a Solaris platform or on a C-series platform.

You can also use SRC configuration applications to configure the SRC software on a Solaris platform. See *Chapter 2, Integrating Third-Party Network Devices into the SRC Network on a Solaris Platform*.

The chapter contains the following topics:

- Overview of Integrating Network Devices into the SRC Network on page 3
- Logging In Subscribers and Creating Sessions on page 5
- Configuration Tasks for Integrating Third-Party Network Devices on page 9
- Setting Up Script Services on page 10
- Adding Objects for Network Devices on page 10
- Setting Up SAE Communities on page 12
- Configuring SAE Properties for the Event Notification API on page 14
- Developing Initialization Scripts for Network Devices on page 15
- Using SNMP to Retrieve Information from Network Devices on page 18
- Using the NIC Resolver on page 18

Overview of Integrating Network Devices into the SRC Network

You can integrate third-party routers and other network devices into your SRC network. The SAE provides a driver that you can use to integrate the SAE with a third-party device. This device driver uses the session store to store and replicate subscriber and service session data within a community of SAEs.

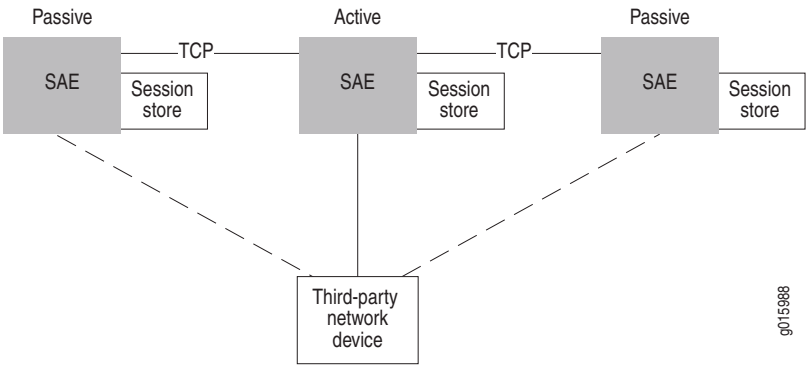
To log in subscribers to the SAE, you use assigned IP subscribers or event notification from an IP address manager.

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

SAE Communities

For SAE redundancy in an SRC network, you can have a community of two or more SAEs. SAEs in a community are given the role of either active SAE or passive SAE. The active SAE manages the connection to the network device and keeps session data up to date within the community. Figure 1 shows a typical SAE community.

Figure 1: SAE Community



When an SAE starts, it negotiates with other SAEs to determine which SAE controls the network device. The SAE community manager and members of the community select the active SAE.

A passive SAE needs to take over as active SAE in any of the following cases:

- The active SAE shuts down. In this case, the active SAE notifies the passive SAEs, and one of the passive SAEs takes over as active SAE.
- A passive SAE does not receive a keepalive message from the active SAE within the keepalive interval. In this case, the passive SAE attempts to become the active SAE.

Storing Session Data

To aid in recovering from an SAE failover, the SAE stores subscriber and service session data. When the SAE manages a network device, session data is stored in the SAE host's file system. The SRC component that controls the storage of session data on the SAE is called the session store. The session store queues data and then writes the data to session store files on the SAE host's disk. Once the data is written to disk, it can survive a server reboot.

For more information, see *Storing Subscriber and Service Session Data* in *SDX Network Guide: SAE, Juniper Networks Routers, and NIC, Chapter 2, Configuring the SAE with the SRC CLI*.

Using Script Services to Provision Third-Party Devices

You use script services to activate services and provision policies on third-party network devices. A script service is a service into which you can insert or reference a script. You write a script that will activate services and provision policies on the third-party device, and then you insert the script into the script service or reference the script in the service. When the SAE activates a service, it runs the script. The script provisions policies on the device using a means that the device supports. You can also include an interface in the script that causes the SAE to send authentication and tracking events when it activates, modifies, or deactivates a script service session.

The SAE core API includes two interfaces for creating a script:

- ScriptService—Defines a service provider interface (SPI) that the script service must implement. The implementation of the ScriptService interface activates, modifies, or deactivates the service.

- ServiceSessionInfo—Provides a callback interface into the SAE and provides information about the service session to the script service.

For information about the ScriptService interface and the ServiceSessionInfo interface, see the script service documentation in the SRC software distribution in the folder *SDK/doc/sae* or in the SAE core API documentation on the Juniper Networks Web site at

http://www.juniper.net/techpubs/software/management/sdx/api-index.html

You can write the script in Java or Jython.

Logging In Subscribers and Creating Sessions

You can use two mechanisms to obtain subscriber address requests and other information and to set up a pseudointerface on the network device. (You must choose one mechanism; you cannot mix them.)

1. Assigned IP subscriber. The SAE learns about a subscriber through subscriber-initiated activities, such as activating a service through the portal or through the SRC SOAP Gateway (SRC-SG).

   With this method, you use the assigned IP subscriber login type along with the network interface collector (NIC) to map IP addresses to the SAE.

2. Event notification from an IP address manager. The SAE learns about subscribers through notifications from an external IP address manager, such as a DHCP server or a RADIUS server.

   With this method, you use the event notification application programming interface (API). The API provides an interface to the IP address manager, and lets the IP address manager notify the SAE of events such as IP address assignments.

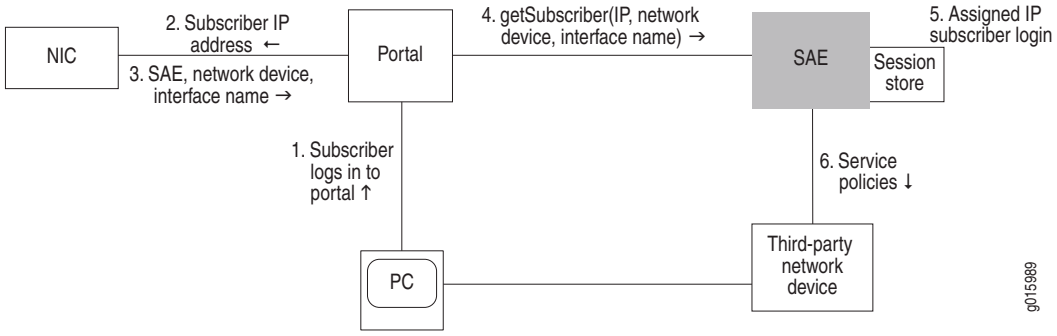
Assigned IP Subscribers

With the assigned IP subscriber method of logging in subscribers and creating sessions, the SRC software uses IP address pools along with network information collector (NIC) resolvers to provide mapping of IP addresses to SAEs. You configure the static address pools or dynamically discovered address pools in the virtual router configuration for a network device. These pools are published in the NIC. The NIC maps subscriber IP addresses in requests received through the portal or SRC-SG to the SAE that currently manages that network device.

Login Interactions with Assigned IP Subscribers

This section describes login interactions for assigned IP subscribers. In the example shown in Figure 2, the subscriber activates a service through a portal. You could also have the subscriber activate a service through the SRC-SG.

Figure 2: Login Interactions with Assigned IP Subscribers



The sequence of events for logging in and creating sessions for assigned IP subscribers is:

1. The subscriber logs in to the portal.
2. The portal sends the subscriber's IP address to the NIC.
3. Based on the IP address, the NIC looks up the subscriber's SAE, network device, and interface name, and returns this information to the portal.
4. The portal sends a getSubscriber message to the SAE. The message includes the subscriber's IP address, network device, and interface name.
5. The SAE creates an assigned IP subscriber and performs a subscriber login. Specifically, it:
    a. Runs the subscriber classification script with the IP address of the subscriber. (Use the ASSIGNEDIP login type in subscriber classification scripts.)
    b. Loads the subscriber profile.
    c. Runs the subscriber authorization plug-ins.

d. Runs the subscriber tracking plug-ins.

e. Creates a subscriber session and stores the session data in the session store file.

6. The SAE pushes service policies for the subscriber session to the network device.

Because the SAE is not notified when the subscriber logs out, the assigned IP idle timer begins when no service is active. The SAE removes the interface subscriber session when the timeout period ends.

Event Notification from an IP Address Manager

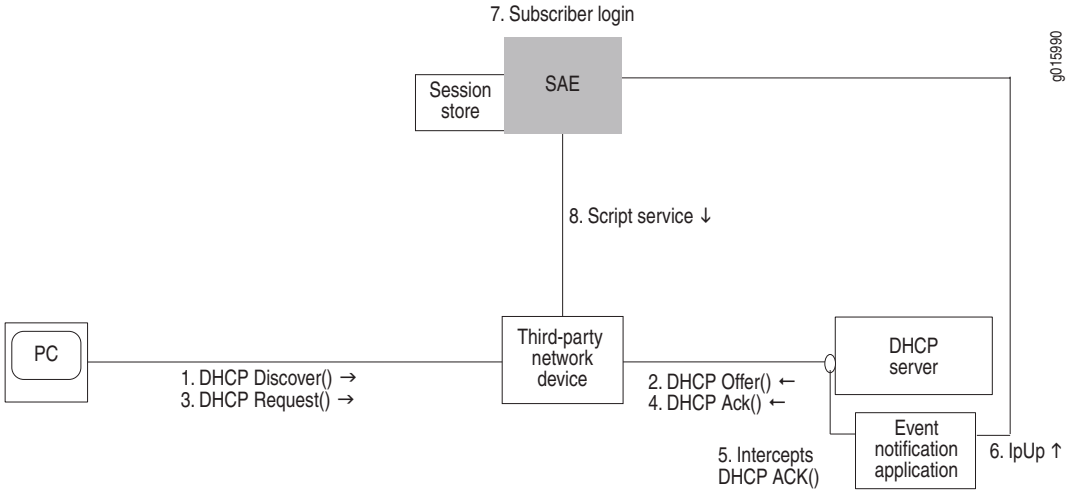
With the event notification method of logging in subscribers and creating subscriber sessions, the subscriber logs in to the network device and obtains an IP address through an address server, usually a DHCP server. The SAE receives notifications about the subscriber, such as the subscriber's IP address, from an event notification application that is installed on the DHCP server.

To use this method of logging in subscribers, you can use the event notification API to create the application that notifies the SAE when events occur between the DHCP server and the network device. You can also use Monitoring Agent, an application that was created with the event notification API and that monitors DHCP or RADIUS messages for DHCP or RADIUS servers. See *SRC Application Library Guide, Chapter 27, Integrating IP Address Managers with the SAE*.

Login with Event Notification

This section describes login interactions by means of event notifications.

Figure 3: Login Interactions with Event Notification Application



The sequence of events for logging in subscribers and creating sessions is:

1. The DHCP client in the subscriber's computer sends a DHCP discover request to the DHCP server.
2. The DHCP server sends a DHCP offer to the subscriber's DHCP client.
3. The DHCP client sends a DHCP request to the DHCP server.
4. The DHCP server acknowledges the request by sending a DHCP Ack message to the DHCP client.
5. The event notification application that is running on the DHCP server intercepts the DHCP Ack message.
6. The event notification application sends an ipUp message to the SAE that notifies the SAE that an IP address is up.
7. The SAE performs a subscriber login. Specifically, it:
    a. Runs the subscriber classification script.
    b. Loads the subscriber profile.
    c. Runs the subscriber authorization plug-ins.
    d. Runs the subscriber tracking plug-ins.
    e. Creates a subscriber session and stores the session in the session store file.
8. The SAE can start script services.

The ipUp event should be sent with a timeout set to the DHCP lease time. The DHCP server sends an ipUp event for each Ack message sent to the client. The SAE restarts the timeout each time it receives an ipUp event.

If the client explicitly releases the DHCP address (that is, it sends a DHCP release event), the DHCP server sends an ipDown event. If the client does not renew the address, the lease expires on the DHCP server and the timeout expires on the SAE.

Configuration Tasks for Integrating Third-Party Network Devices

To integrate third-party devices into your SRC network, complete the following tasks:

- Write a script and add a script service that references the script.

  See *Setting Up Script Services* on page 10.

- Add objects for the devices.

  See *Adding Objects for Network Devices* on page 10.

- Configure an SAE community.

  See *Setting Up SAE Communities* on page 12.

- (Optional) Configure SAE properties for the Event Notification API if you are using the event notification method to log in subscribers.

  See *Configuring SAE Properties for the Event Notification API* on page 14.

- Configure the session store.

  See *Storing Subscriber and Service Session Data* in *SDX Network Guide: SAE, Juniper Networks Routers, and NIC, Chapter 2, Configuring the SAE with the SRC CLI*.

- If you are using the event notification method to log in subscribers, integrate the SAE with an IP address manager. There are two ways to do so:

  - Use the event notification API to create an application that notifies the SAE when events occur between the DHCP server and the network device.

    See the event notification API documentation in the SRC software distribution in the folder *SDK/doc/sae* or in the SAE CORBA remote API documentation on the Juniper Networks Web site at

    http://www.juniper.net/techpubs/software/management/sdx/api-index.html

  - Use Monitoring Agent, an application that was created with the event notification API and that monitors DHCP or RADIUS messages for DHCP or RADIUS servers.

    See *SRC Application Library Guide, Chapter 27, Integrating IP Address Managers with the SAE*.

Setting Up Script Services

To set up script services:

1. Write a script that implements the ScriptService interface, a service provider interface (SPI) for the SAE.

   See *SDX Services and Policies Guide, Chapter 1, Managing Services with the SRC CLI*.

   See the script service documentation in the SRC software distribution in the folder *SDK/doc/sae* or in the SAE core API documentation on the Juniper Networks Web site at

   http://www.juniper.net/techpubs/software/management/sdx/api-index.html

2. Add a script service that references the script.

   See *SDX Services and Policies Guide, Chapter 1, Managing Services with the SRC CLI*.

Adding Objects for Network Devices

For each network device that the SAE manages, add a router object and virtual router object.

Use the following configuration statements to add a router object:

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

To add a router object:

1. From configuration mode, access the configuration statements that configure network devices. This sample procedure uses proxy_device as the name of the router.

   user@host# **edit shared network device proxy_device**

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

   [edit shared network device proxy_device]
   user@host# **set description** *description*

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

   [edit shared network device proxy_device]
   user@host# **set management-address** *management-address*

4. Set the type of device that you are adding to proxy.

   [edit shared network device proxy_device]
   user@host# **set device-type proxy**

5. (Optional) Verify your configuration.

```
[edit shared network device proxy_device]
user@host# show
description "Third-party router";
management-address 192.168.9.25;
device-type proxy;
interface-classifier {
  rule rule-0 {
    script #;
  }
}
```

Adding Virtual Router Objects

Use the following configuration statements to add a virtual router:

shared network device name virtual-router *name* {
    sae-connection [*sae-connection*...];
    snmp-read-community *snmp-read-community*;
    snmp-write-community *snmp-write-community*;
    scope [*scope*...];
    tracking-plug-in [*tracking-plug-in*...];
}

To add a virtual router:

1. From configuration mode, access the configuration statements for virtual routers. This sample procedure uses proxy_device as the name of the router object. For third-party devices, use the name default for the virtual router.

   user@host# **edit shared network device proxy_device virtual-router default**

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

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

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

   [edit shared network device proxy_device virtual-router default]
   user@host# **set snmp-read-community** *snmp-read-community*

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

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

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

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

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

   [edit shared network device proxy_device virtual-router default]
   user@host# **set tracking-plug-in** [*tracking-plug-in*...]

7. (Optional) Verify your configuration.

```
[edit shared network device proxy_device virtual-router default]
user@host# show
sae-connection 10.8.221.45;
snmp-read-community ********;
snmp-write-community ********;
scope POP-Toronto;
tracking-plug-in flexRadius;
```

Setting Up SAE Communities

You can configure the following for SAE communities:

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

  See *Adding Virtual Router Objects* on page 11.

- Configure parameters for the SAE community manager.

  See *Configuring the SAE Community Manager* on page 12.

- Specify the name of the community manager.

  See *Specifying the Community Manager in the SAE Device Driver* on page 14.

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

Configuring the SAE Community Manager

Use the following configuration statements to configure the SAE community manager that manages third-party network device communities:

shared sae configuration external-interface-features *name* CommunityManager {
    keepalive-interval *keepalive-interval*;
    threads *threads*;
    acquire-timeout *acquire-timeout*;
    blackout-time *blackout-time*;
}

To configure the community manager:

1. From configuration mode, access the configuration statements for the community manager. In this sample procedure, sae_mgr is the name of the community manager.

   user@host# **edit shared sae configuration external-interface-features sae_mgr CommunityManager**

2. Specify the interval between keepalive messages sent from the active SAE to the passive members of the community.

   [edit shared sae configuration external-interface-features sae_mgr CommunityManager]
   user@host# **set keepalive-interval** *keepalive-interval*

3. Specify the number of threads that are allocated to manage the community. You generally do not need to change this value.

   [edit shared sae configuration external-interface-features sae_mgr CommunityManager]
   user@host# **set threads** *threads*

4. Specify the amount of time an SAE waits for a remote member of the community when it is acquiring a distributed lock. You generally do not need to change this value.

   [edit shared sae configuration external-interface-features sae_mgr CommunityManager]
   user@host# **set acquire-timeout** *acquire-timeout*

5. Specify the amount of time that an active SAE must wait after it shuts down before it can try to become the active SAE of the community again.

   [edit shared sae configuration external-interface-features sae_mgr CommunityManager]
   user@host# **set blackout-time** *blackout-time*

6. (Optional) Verify the configuration of the SAE community manager.

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

Specifying the Community Manager in the SAE Device Driver

Use the following configuration statements to specify the community manager in the SAE device driver.

```
shared sae configuration driver third-party {
    sae-community-manager sae-community-manager;
}
```

To specify the community manager:

1. From configuration mode, access the configuration statements for the third-party device driver.

   ```
   user@host# edit shared sae configuration driver third-party
   ```

2. Specify the name of the community manager.

   ```
   [edit shared sae configuration driver third-party]
   user@host# set sae-community-manager sae-community-manager
   ```

3. (Optional) Verify the configuration of the third-party device driver.

   ```
   [edit shared sae configuration driver third-party]
   user@host# show
   sae-community-manager sae_mgr;
   ```

Configuring SAE Properties for the Event Notification API

Use the following configuration statements to configure properties for the event notification API:

```
shared sae configuration external-interface-features name EventAPI {
    retry-time retry-time;
    retry-limit retry-limit;
    threads threads;
}
```

To configure properties for the event notification API:

1. From configuration mode, access the configuration statements for the event notification API. In this sample procedure, event_api is the name of the Event API configuration.

   ```
   user@host# edit shared sae configuration external-interface-features event_api
   EventAPI
   ```

2. Specify the amount of time between attempts to send events that could not be delivered.

   ```
   [edit shared sae configuration external-interface-features event_api EventAPI]
   user@host# set retry-time retry-time
   ```

3. Specify the number of times an event fails to be delivered before the event is discarded.

   [edit shared sae configuration external-interface-features event_api EventAPI]
   user@host# **set retry-limit** *retry-limit*

4. Specify the number of threads allocated to process events.

   [edit shared sae configuration external-interface-features event_api EventAPI]
   user@host# **set threads** *threads*

5. (Optional) Verify the configuration of the event notification API properties.

```
[edit shared sae configuration external-interface-features event_api
EventAPI]
user@host# show
EventAPI {
  retry-time 300;
  retry-limit 5;
  threads 5;
}
```

Developing Initialization Scripts for Network Devices

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

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

Interface Object Fields

Scripts for network devices interact with the SAE through an interface object called Ssp. The SAE exports a number of fields through the interface object to the script and expects the script to provide the entry point to the SAE.

Table 4 describes the fields that the SAE exports.

Table 4: Exported Fields

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

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

- <VRName>—Name of the virtual router object that has been configured for the network device in the format: virtualRouterName@RouterName
- <virtualIp>—Virtual IP address of the SAE (string, dotted decimal; for example: 192.168.254.1)
- <realIp>—Real IP address of the SAE (string, dotted decimal; for example, 192.168.1.20)
- <VRIp>—IP address of the virtual router (string, dotted decimal)
- <transportVR>—Name of the virtual router

The factory function must implement the following interface:

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

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

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

Required Methods

Instances of the interface object must implement the following methods:

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

Example: Initialization Script

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

```
class SillyRouterInit:
    def __init__(self, vrName, virtualIp, realIp, vrIp, transportVr):
        """ initialize router initialization object """
        self.vrName = vrName
        Ssp.infoLog.println("SillyRouterInit created")
```

```
    def setup(self):
        """ initialize connection to router """
        Ssp.infoLog.printin("Setup connection to VR %(vrName)s" %
                    vars(self))

    def shutdown(self):
        """ shutdown connection to router """
        Ssp.infoLog.printin("Shutdown connection to VR %(vrName)s" %
                    vars(self))

#
# publish interface object to Ssp core
#
Ssp.routerInit = SillyRouterInit
```

Copying Initialization Scripts to the C-series Platform

If you use a script that is not provided with the SRC software, you need to use the file copy command to copy your script to the C-series platform. For example:

user@host> **file copy ftp://user@myserver/routerinit.py /opt/UMC/sae/lib**
Password:

Specifying Initialization Scripts on the SAE

Use the following configuration statements to specify initialization scripts for third-party devices:

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

To configure initialization scripts for third-party devices:

1. From configuration mode, access the configuration statements that configure initialization scripts.

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

2. Specify the initialization script for third-party devices.

   [edit shared sae configuration driver scripts]
   user@host# **set general** *general*

3. Configure a path to scripts that are not in the default location, */opt/UMC/sae/lib*.

   [edit shared sae configuration driver scripts]
   user@host# **set extension-path** *extension-path*

4. (Optional) Verify your initialization script configuration.

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

Using SNMP to Retrieve Information from Network Devices

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

We recommend that you specify SNMP communities for each virtual router object. (See *Adding Virtual Router Objects* on page 11.) You can also configure global default SNMP communities.

Configuring Global SNMP Communities in the SRC Software

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

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

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

To configure global default SNMP communities:

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

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

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

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

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

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

Using the NIC Resolver

If you are using the assigned IP subscriber method of logging in subscribers, and you are using the NIC to determine the subscriber's SAE, you need to configure a resolver on the NIC. The OnePopDynamicIp sample configuration data supports this scenario. The OnePopDynamicIp configuration supports one point of presence (POP) and provides no redundancy. The realm for this configuration accommodates the situation in which IP pools are configured locally on each virtual router object.

You can access the OnePopDynamicIp configuration in the SRC CLI. See *SDX Network Guide: SAE, Juniper Networks Routers, and NIC, Chapter 10, Configuring NIC with the SRC CLI* for information about configuring NIC scenarios.

Chapter 2

Integrating Third-Party Network Devices into the SRC Network on a Solaris Platform

This chapter describes how to integrate third-party network devices into the SRC network with the SRC configuration applications that run only on Solaris platforms.

You can also use the CLI that runs on Solaris platforms and the C-series platform to configure the SRC system to work with third-party devices. See *Chapter 1, Integrating Third-Party Network Devices into the SRC Network with the SRC CLI*.

The chapter contains the following topics:

- Overview of Integrating Network Devices into the SRC Network on page 21
- Logging In Subscribers and Creating Sessions on page 23
- Configuration Tasks for Integrating Third-Party Network Devices on page 27
- Setting Up Script Services on page 28
- Adding Objects for Network Devices to the Directory on page 28
- Setting Up SAE Communities on page 34
- Configuring SAE Properties for the Event Notification API on page 38
- Developing Initialization Scripts for Network Devices on page 39
- Using SNMP to Retrieve Information from Network Devices on page 43
- Using the NIC Resolver on page 44

Overview of Integrating Network Devices into the SRC Network

You can integrate third-party routers and other network devices into your SRC network. The SAE provides a driver that you can use to integrate the SAE with a third-party device. This device driver uses the session store to store and replicate subscriber and service session data within a community of SAEs.

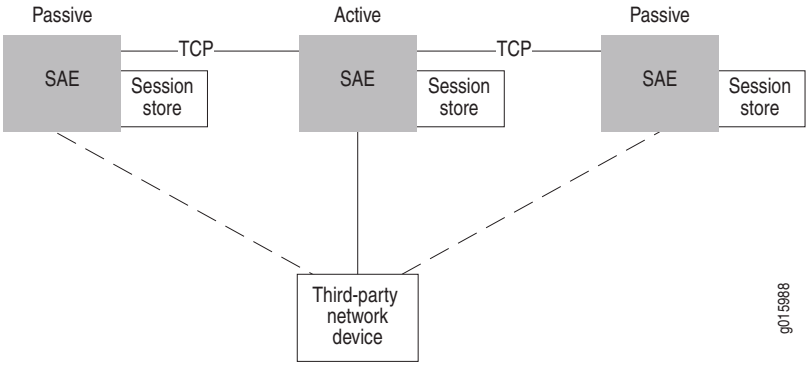
To log in subscribers to the SAE, you use assigned IP subscribers or event notification from an IP subscribers.

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

SAE Communities

For SAE redundancy in an SRC network, you can have a community of two or more SAEs. SAEs in a community are given the role of either active SAE or passive SAE. The active SAE manages the connection to the network device and keeps session data up to date within the community. Figure 4 shows a typical SAE community.

Figure 4: SAE Community



When an SAE starts, it negotiates with other SAEs to determine which SAE controls the network device. The SAE community manager and members of the community select the active SAE.

A passive SAE needs to take over as active SAE in any of the following cases:

- The active SAE shuts down. In this case, the active SAE notifies the passive SAEs, and one of the passive SAEs takes over as active SAE.
- A passive SAE does not receive a keepalive message from the active SAE within the keepalive interval. In this case, the passive SAE attempts to become the active SAE.

Storing Session Data

To aid in recovering from an SAE failover, the SAE stores subscriber and service session data. When the SAE manages a network device, session data is stored locally in the SAE host's file system. The SRC component that controls the storage of session data on the SAE is called the session store. The session store queues data and then writes the data to session store files on the SAE host's disk. Once the data is written to disk, it can survive a server reboot.

For more information, see *Storing Subscriber and Service Session Data* in *SDX Network Guide: SAE, Juniper Networks Routers, and NIC, Chapter 3, Configuring the SAE with SDX Configuration Editor*.

Using Script Services to Provision Third-Party Devices

You use script services to activate services and provision policies on third-party network devices. A script service is a service into which you can insert or reference a script. You write a script that will activate services and provision policies on the third-party device, and then you insert the script into the script service or reference the script in the service. When the SAE activates a service, it sends the script to the network device that the SAE is managing. You can also include an interface in the script that will cause the SAE to send authentication and tracking events when it activates, modifies, or deactivates a script service session.

The SAE core API includes two interfaces for creating a script:

- ScriptService—Defines a service provider interface (SPI) that the script service must implement. The implementation of the ScriptService interface activates the service.
- ServiceSessionInfo—Provides a callback interface into SAE and provides information about the service session to the script service.

For information about the ScriptService interface and the ServiceSessionInfo interface, see the script service documentation in the SRC software distribution in the folder *SDK/doc/sae* or in the SAE core API documentation on the Juniper Networks Web site at

http://www.juniper.net/techpubs/software/management/sdx/api-index.html

You can write the script in Java or Jython.

Logging In Subscribers and Creating Sessions

You can use two mechanisms to obtain subscriber address requests and other information and to set up a pseudointerface on the network device. (You must choose one mechanism; you cannot mix them.):

1. Assigned IP subscriber. The SAE learns about a subscriber through subscriber-initiated activities, such as activating a service through the portal or through the SRC SOAP Gateway (SRC-SG).

   With this method, you use the assigned IP subscriber login type along with the network interface collector (NIC) to map IP addresses to the SAE.

2. Event notification from an IP address manager. The SAE learns about subscribers through notifications from an external IP address manager, such as a DHCP server or a RADIUS server.

   With this method, you use the event notification application programming interface (API). The API provides an interface to the IP address manager, and lets the IP address manager notify the SAE of events such as IP address assignments.

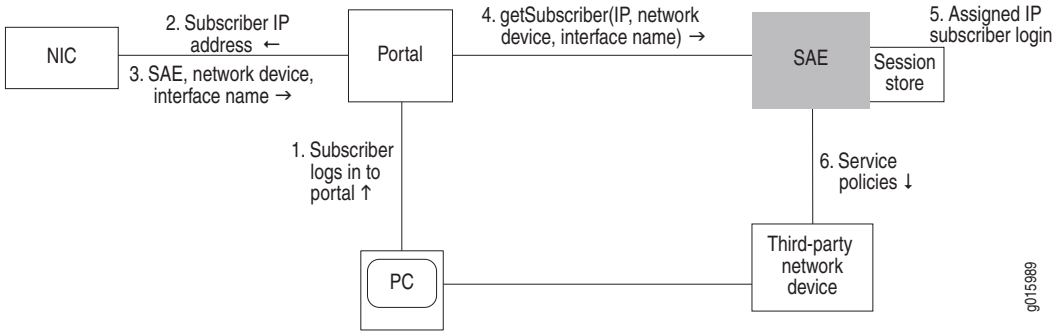
Assigned IP Subscribers

With the assigned IP subscriber method of logging in subscribers and creating sessions, the SRC software uses IP address pools along with NIC resolvers to provide mapping of IP addresses to SAEs. You configure the static address pools or dynamically discovered address pools in the virtual router configuration for a network device. These pools are published in the NIC. The NIC maps subscriber IP addresses in requests received through the portal or Advanced Services Gateway to the SAE that currently manages that network device.

Login Interactions with Assigned IP Subscribers

This section describes login interactions for assigned IP subscribers. In the example shown in Figure 5, the subscriber activates a service through a portal. You could also have the subscriber activate a service through the Advanced Services Gateway.

Figure 5: Login Interactions with Assigned IP Subscribers



The sequence of events for logging in and creating sessions for assigned IP subscribers is:

1. The subscriber logs in to the portal.
2. The portal sends the subscriber's IP address to the NIC.
3. Based on the IP address, the NIC looks up the subscriber's SAE, network device, and interface name, and returns this information to the portal.
4. The portal sends a getSubscriber message to the SAE. The message includes the subscriber's IP address, network device, and interface name.
5. The SAE creates an assigned IP subscriber and performs a subscriber login. Specifically, it
    a. Runs the subscriber classification script with the IP address of the subscriber. (Use the ASSIGNEDIP login type in subscriber classification scripts.)
    b. Loads the subscriber profile.
    c. Runs the subscriber authorization plug-ins.

d. Runs the subscriber tracking plug-ins.

e. Creates a subscriber session and stores the session data in the session store file.

6. The SAE pushes service policies for the subscriber session to the network device.

Because the SAE is not notified when the subscriber logs out, the assigned IP idle timer begins when no service is active. The SAE removes the interface subscriber session when the timeout period ends.

Event Notification from an IP Address Manager

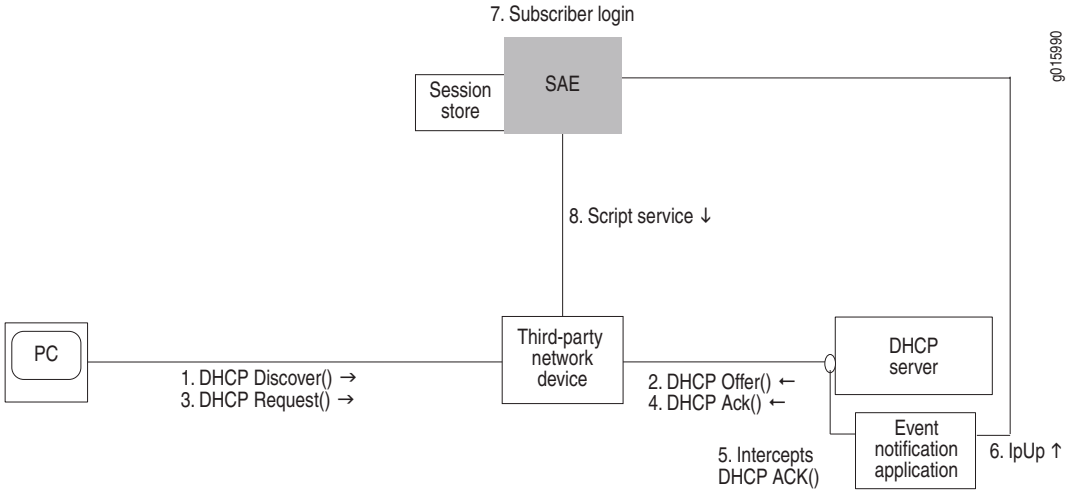
With the event notification method of logging in subscribers and creating subscriber sessions, the subscriber logs in to the network device and obtains an IP address through an address server, usually a DHCP server. The SAE receives notifications about the subscriber, such as the subscriber's IP address, from an event notification application that is installed on the DHCP server.

To use this method of logging in subscribers, you can use the event notification API to create the application that notifies the SAE when events occur between the DHCP server and the network device. You can also use Monitoring Agent, an application that was created with the event notification API, and that monitors DHCP or RADIUS messages for DHCP or RADIUS servers. See *SRC Application Library Guide, Chapter 27, Integrating IP Address Managers with the SAE*.

Login with Event Notification

This section describes login interactions using event notifications.

Figure 6: Login Interactions with Event Notification Application



The sequence of events for logging in subscribers and creating sessions is:

1. The DHCP client in the subscriber's computer sends a DHCP discover request to the DHCP server.
2. The DHCP server sends a DHCP offer to the subscriber's DHCP client.
3. The DHCP client sends a DHCP request to the DHCP server.
4. The DHCP server acknowledges the request by sending a DHCP Ack message to the DHCP client.
5. The event notification application that is running on the DHCP server intercepts the DHCP Ack message.
6. The event notification application sends an ipUp message to the SAE that notifies the SAE that an IP address is up.
7. The SAE performs a subscriber login. Specifically, it:
    a. Runs the subscriber classification script.
    b. Loads the subscriber profile.
    c. Runs the subscriber authorization plug-ins.
    d. Runs the subscriber tracking plug-ins.
    e. Creates a subscriber session and stores the session in the session store file.
8. The SAE can start script services.

The ipUp event should be sent with a timeout set to the DHCP lease time. The DHCP server sends an ipUp event for each Ack sent to the client. The SAE restarts the timeout each time it receives an ipUp event.

If the client explicitly releases the DHCP address (that is, it sends a DHCP release event), the DHCP server sends an ipDown event. If the client does not renew the address, the lease expires on the DHCP server and the timeout expires on the SAE.

Configuration Tasks for Integrating Third-Party Network Devices

To integrate third-party devices into your SRC network, you need to complete the following tasks:

- Write a script and add a script service that references the script.

  See *Setting Up Script Services* on page 28.

- Add objects for the devices to the directory.

  Adding Objects for Network Devices to the Directory on page 28.

- Set up an SAE community.

  See *Setting Up SAE Communities* on page 34.

- Configure SAE properties for the Event Notification API.

  See *Configuring SAE Properties for the Event Notification API* on page 38.

- Configure the session store.

  See *Storing Subscriber and Service Session Data* on page 41 in *SDX Network Guide: SAE, Juniper Networks Routers, and NIC, Chapter 3, Configuring the SAE with SDX Configuration Editor*.

- If you are using the event notification method to log in subscribers, integrate the SAE with an IP address manager. There are two ways to do so:

  - Use the event notification API to create an application that notifies the SAE when events occur between the DHCP server and the network device.

    See the event notification API documentation in the SRC software distribution in the folder *SDK/doc/sae* or in the SAE CORBA remote API documentation on the Juniper Networks Web site at

    http://www.juniper.net/techpubs/software/management/sdx/api-index.html

  - Use Monitoring Agent, an application that was created with the event notification API, and that monitors DHCP or RADIUS messages for DHCP or RADIUS servers.

    See *SRC Application Library Guide, Chapter 27, Integrating IP Address Managers with the SAE*.

Setting Up Script Services

To set up script services:

1. Write a script that implements the ScriptService interface, a service provider interface (SPI) for the SAE.

   See *SDX Services and Policies Guide, Chapter 2, Managing Services on a Solaris Platform*.

   See the script service documentation in the SRC software distribution in the folder *SDK/doc/sae* or in the SAE core API documentation on the Juniper Networks Web site at

   http://www.juniper.net/techpubs/software/management/sdx/api-index.html

2. Add a script service that references the script.

   See *SDX Services and Policies Guide, Chapter 2, Managing Services on a Solaris Platform*.

Adding Objects for Network Devices to the Directory

For each network device that the SAE manages, you need to add a router object and virtual router object to the directory.

To add a router object to the directory with SDX Admin:

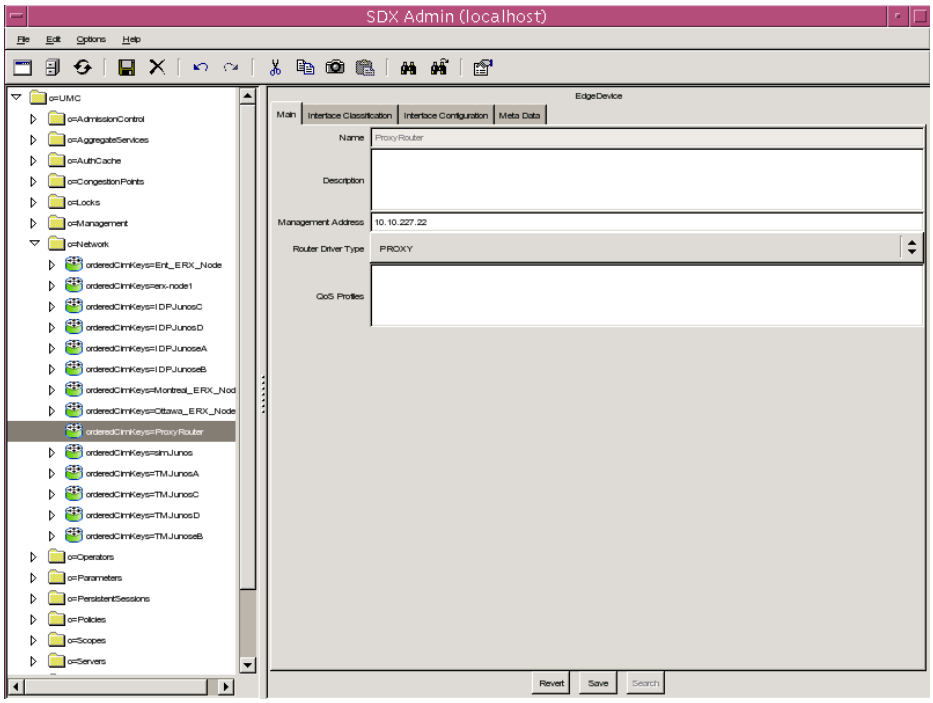
1. In the navigation pane, highlight *o = Network*, and right-click.

2. Select **New > EdgeDevice**.

   The New EdgeDevice dialog box appears.

3. In the New EdgeDevice dialog box, enter a name for the network device, and click **OK**.

The name of the new device appears in the navigation pane.



4. Use the field descriptions in *Router Fields* on page 29 to configure the router, and then click **Save**.

Router Fields

Use the fields in this section to configure routers.

Description

- Information about this device; keywords that the SDX find utility uses.
- Value—Text string
- Example—ERX-1400 router located in Ottawa

Management Address

- IP address of the network device.
- Value—IP address
- Example—192.0.1.1

Router Driver Type

- Type of device that this directory object will be used to manage.
- Value—Select PROXY
- Default—No value

QoS Profiles

- This field applies to JUNOSe routers only

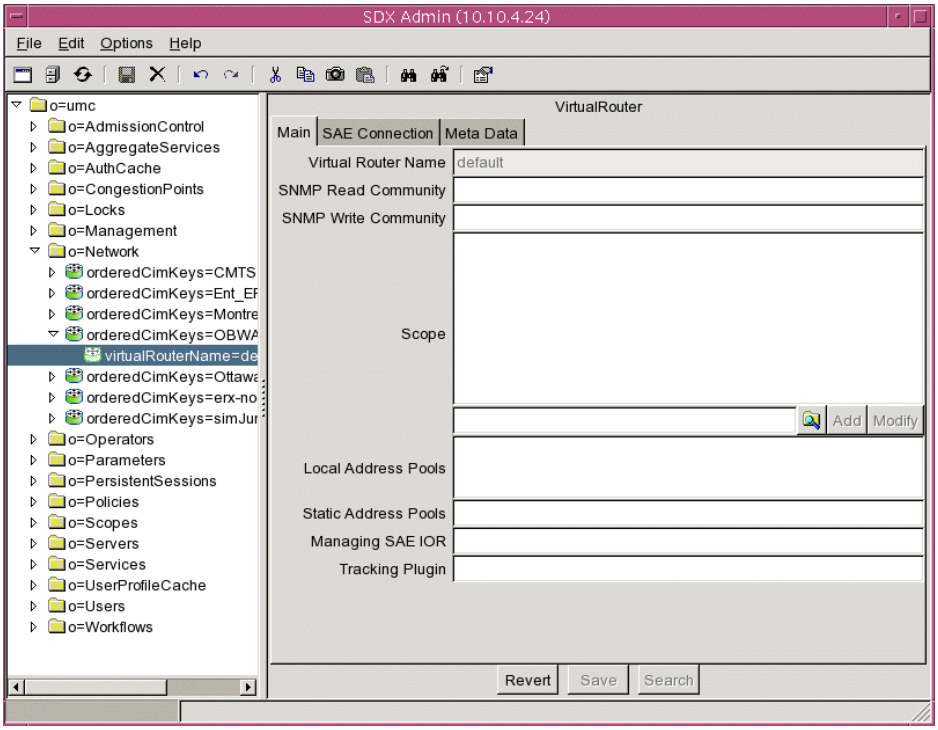
Adding Virtual Router Objects

To add a virtual router object to the directory with SDX Admin:

1. In the navigation pane, highlight the device to which you want to add the VR, and right-click.
2. Select **New > VirtualRouter**.

   The New VirtualRouter dialog box appears.
3. Enter the name default, and click **OK**.

   The name of the new virtual router object appears in the navigation pane, and the VirtualRouter pane appears.



4. Set the parameters in the Main tab in the VirtualRouter pane. See *Virtual Router Fields* on page 31.

5. Select the SAE Connection tab in the VirtualRouter pane, and add SAEs that are connected to the router. See *Defining SAE Communities* on page 34.

NOTE: This step is required for the SAE to work with the router.

6. Click Save in the VirtualRouter pane.

Virtual Router Fields

Use the fields in this section to define virtual routers.

SNMP Read Community

- SNMP community name associated with SNMP read-only operations for this VR.
- Value—Text string
- Example—admin

SNMP Write Community

- SNMP community name associated with SNMP write operations for this VR.
- Value—Text string
- Example—public

Scope

- Service scopes assigned to this VR—See *SDX Services and Policies Guide, Chapter 2, Managing Services on a Solaris Platform*.
- Value—Text string
- Example—POP-Westford

Local Address Pools

- For JUNOSe routers only. List of IP address pools that a JUNOSe VR currently manages and stores.
- Value—You can specify an unlimited number of ranges of local IP address pools for JUNOSe VRs. You can specify either the first and last addresses in a range or the first IP address and a factor that indicates the start of the range. You can also specify IP addresses to exclude. Use spaces in the syntax only to separate the first and last explicit IP addresses in a range.

  The IP pool syntax has the format:

  ([<ipAddressStart> <ipAddressEnd>] |
  {<ipBaseAddress>/(<mask> | <digitNumber>)(,<ipAddressExclude>)*})

  - < ipAddressStart > —First IP address (version 4 or 6) in a range
  - < ipAddressEnd > —Last IP address (version 4 or 6) in a range
  - < ipBaseAddress > —Network base address

- <mask>—IP address mask
- <digitNumber>—Integer specifying the number of significant digits of the first IP address in the range
- <ipAddressExclude>—List of IP addresses to be excluded from the range
- |—Choice of expression; choose either the expression to the left or the expression to the right of this symbol
- *—Zero or more instances of the preceding group

- Guidelines—Configure this field on JUNOSe VRs only. If you do not configure the **PoolPublisher** router initialization scripts for a JUNOSe router, configure this field for the JUNOSe VR.
- Default—No value
- Example—This example shows four ranges for the IP address pool.

  ([10.10.10.5 10.10.10.250]
  {10.20.20.0/24}
  {10.21.0.0/255.255.0.0}
  {10.20.30.0/24,10.20.30.1})

  - The first range (a simple range) specifies all the IP addresses between the two IP addresses 10.10.10.5 and 10.10.10.250.
  - The second range specifies all the IP addresses in the range 10.20.20.0 to 10.20.20.255.
  - The third range uses a network mask to specify all the IP addresses in the range 10.21.0.0 to 10.21.255.255.
  - The fourth range specifies all the addresses of the network 10.20.30.0 to 10.20.30.255, excluding the address 10.20.30.1.

Static Address Pools

- For JUNOSe routers and CMTS devices only. List of IP address pools that a JUNOSe VR manages but does not store. You can configure these address pools only in the SRC software.
- Value—See the field Local Address Pools.
- Guidelines—Configure this field on JUNOSe and CMTS VRs only.
- Default—No value
- Example—([10.10.10.5 10.10.10.250] {10.20.20.0/24})

Managing SAE IOR

- Common Object Request Broker Architecture (CORBA) reference for the SAE managing this VR.
- Value—One of the following items:
  - The actual CORBA reference for the SAE
  - The absolute path to the interoperable object reference (IOR) file
  - A corbaloc URL in the form corbaloc:: < host > :8801/SAE
    - < host > is the name or IP address of the SAE host.
- Default—No value
- Guidelines—The **PoolPublisher** and **IorPublisher** router initialization scripts provide this information when the router connects to the SAE. If you do not select one of these router initialization scripts, enter a value in this field.
- Example—One of the following items:
  - Absolute path—*/opt/UMC/sae/var/run/sae.ior*
  - corbaloc URL—corbaloc::boston:8801/SAE
  - Actual IOR—IOR:000000000000002438444C3A736D67742E6A756E697...

Tracking Plug-in

- Plug-ins that track interfaces that the SAE manages on this VR. The SAE calls these plug-in instances for every interface it manages. The SAE calls these plug-ins after an interface comes up, when new policies are installed on the interface, and when the interface goes down.
- Value—Comma-separated list of plug-in instances
- Guidelines—Enter plug-in instances and network information collector (NIC) SAE plug-in agents that are specific to this VR.
- Default—No value
- Example—nicsae, flexRadius

Setting Up SAE Communities

You can configure the following for SAE communities:

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

  See *Defining SAE Communities* on page 34.

- Configure parameters for the SAE community manager in SDX Configuration Editor.

  See *Configuring the SAE Community Manager* on page 36.

- Specify the name of the community manager.

  See *Specifying the SAE Community Manager* on page 37.

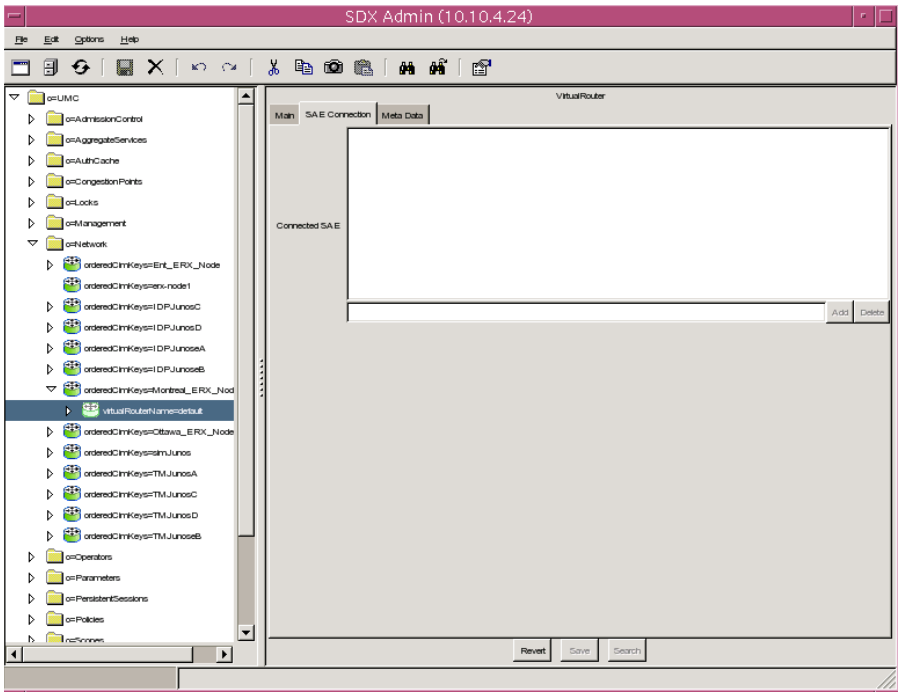
- If there is a firewall in the network, configure the firewall to allow SAE messages through.

Defining SAE Communities

You define SAE communities by entering the SAEs in a community in the connected SAE field of the virtual router object.

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

To define a community, select the SAE Connection tab of the VirtualRouter pane in SDX Admin, and add the addresses of SAEs that can manage the device.



Adding an SAE

To add an SAE:

1. Type the IP address of the SAE in the field below the Connected SAE box.
2. Click **Add**.

Modifying an SAE Address

To modify an SAE address:

1. Click the IP address of the SAE in the Connected SAE box.
2. Modify the IP address in the field below the Connected SAE box.
3. Click **Modify**.

Deleting an SAE Address

To delete an SAE address:

1. Click the IP address of the SAE in the Connected SAE box.
2. Remove the IP address from the field below the Connected SAE box.
3. Click **Delete**.

Connected SAE

- SAEs that are connected to the network device.
- Value—IP addresses
- Default—No value

Configuring the SAE Community Manager

To use SDX Configuration Editor to configure the SAE community manager that manages third-party network device communities:

1. In the navigation pane, select a configuration file for the SAE that you want to configure.
2. Select the Ext. Interface tab, and expand the Community Manager (PROXYCommunityManager) section.

3. Edit or accept the default values in the field.

   See *Community Manager Fields* on page 36.

4. Select **File > Save**.
5. Right-click the configuration file, and select **SDX System Configuration > Export to LDAP Directory**.

Community Manager Fields

In SDX Configuration Editor, you can modify the following field in the Community Manager (PROXYCommunityManager) section of the Ext. Interface pane in an SAE configuration file.

Keepalive Interval [s]

- Interval between keepalive messages sent from the active SAE to the passive members of the community.
- Value—Number of seconds in the range 0–2147483647
- Default—30
- Property name—SAEFeature.PROXYCommunityManager.heartbeat

Number of Threads

- Number of threads that are allocated to manage the community.
- Value—Integer in the range 0–2147483647
- Guidelines—You generally do not need to change this property.
- Default—5
- Property name—SAEFeature.PROXYCommunityManager.num_threads

Acquire Timeout

- Amount of time an SAE waits for a remote member of the community when it is acquiring a distributed lock. To avoid race conditions when the SAE community is determining which SAE is the active SAE, the community manager has a distributed lock. When an SAE attempts to become the active SAE, it needs to acquire the distributed lock.
- Value—Number of seconds in the range 0–2147483647
- Guidelines—You generally do not need to change this property.
- Default—15
- Property name—SAEFeature.PROXYCommunityManager.acquire_timeout

Blackout Time

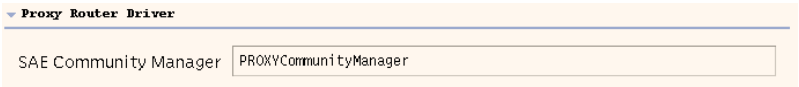
- Amount of time that an active SAE must wait after it shuts down before it can try to become the active SAE of the community again.
- Value—Number of seconds in the range 0–2147483647
- Default—30
- Property name—SAEFeature.PROXYCommunityManager.blackout_time

Specifying the SAE Community Manager

To use SDX Configuration Editor to specify the name of the community manager that manages third-party network device communities:

1. In the navigation pane, select a configuration file for the SAE that you want to configure.

2. Select the Router tab, and expand the Proxy Router Driver section.



3. Edit or accept the default values in the field.

   See *Proxy Router Driver Fields* on page 38.

4. Select **File > Save**.

5. Right-click the configuration file, and **select SDX System Configuration > Export to LDAP Directory**.

Proxy Router Driver Fields

In SDX Configuration Editor, you can modify the following field in the Proxy Router Driver section of the Router pane in an SAE configuration file.

SAE Community Manager

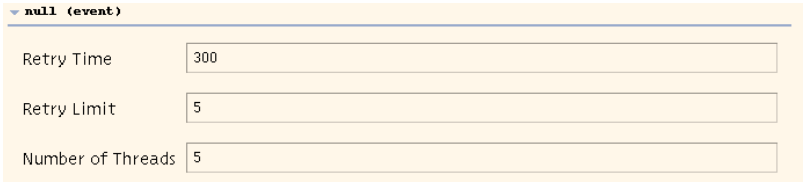
- Name of the community manager that manages network device communities. Active SAEs are selected from this community. You define community managers in the Ext. Interfaces tab of SDX Configuration Editor. See *Configuring the SAE Community Manager* on page 36.
- Value—Community name
- Default—PROXYCommunityManager
- Property name—Router.proxy.community.name

Configuring SAE Properties for the Event Notification API

To use SDX Configuration Editor to specify the name of the community manager that manages third-party network device communities:

1. In the navigation pane, select a configuration file for the SAE that you want to configure.

2. Select the Ext. Interface tab, and expand the Proxy Router Driver section.



3. Edit or accept the default values in the field.

   See *Event API Fields* on page 39.

4. Select **File > Save**.

5. Right-click the configuration file, select **SDX System Configuration > Export to LDAP Directory**.

Event API Fields

In SDX Configuration Editor, you can modify the following fields in the Event API section of the Ext. Interface pane in an SAE configuration file.

Retry Time

- Amount of time between attempts to send events that could not be delivered.
- Value—Number of seconds in the range 0–2147483647
- Default—300
- Property name—SAEFeature.event.retry_time

Retry Limit

- Number of times an event fails to be delivered before the event is discarded.
- Value—Integer in the range 0–2147483647
- Default—5
- Property name—SAEFeature.event.retry_limit

Number of Threads

- Number of threads allocated to process events.
- Value—Integer in the range 0–2147483647
- Default—5
- Property name—SAEFeature.event.num_threads

Developing Initialization Scripts for Network Devices

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

We provide the IorPublisher script in the */opt/UMC/sae/lib* folder. The IorPublisher script publishes the IOR of the SAE in the directory so that a NIC can associate a router with an SAE.

Interface Object Fields

Scripts for network devices interact with the SAE through an interface object called Ssp. The SAE exports a number of fields through the interface object to the script and expects the script to provide the entry point to the SAE.

Table 5 describes the fields that the SAE exports.

Table 5: Exported Fields

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

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

- < VRName > —Name of the virtual router object that has been configured for the network device in the format: virtualRouterName@RouterName
- < virtualIp > —Virtual IP address of the SAE (string, dotted decimal; for example: 192.168.254.1)
- < realIp > —Real IP address of the SAE (string, dotted decimal; for example, 192.168.1.20)
- < VRIp > —IP address of the virtual router (string, dotted decimal)
- < transportVR > —Name of the virtual router

The factory function must implement the following interface:

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

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

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

Required Methods

Instances of the interface object must implement the following methods:

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

Example: Router Initialization Script

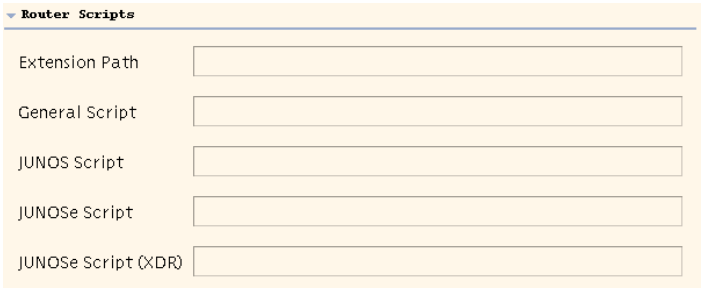
The following script defines a router initialization class named *SillyRouterInit*. The interface class does not implement any useful functionality. The interface class just writes messages to the infoLog when the router connection is created or terminated.

```
class SillyRouterInit:
    def __init__(self, vrName, virtualIp, realIp, vrIp, transportVr):
        """ initialize router initialization object """
        self.vrName = vrName
        Ssp.infoLog.printin("SillyRouterInit created")

    def setup(self):
        """ initialize connection to router """
        Ssp.infoLog.printin("Setup connection to VR %(vrName)s" %
                    vars(self))

    def shutdown(self):
        """ shutdown connection to router """
        Ssp.infoLog.printin("Shutdown connection to VR %(vrName)s" %
                    vars(self))

#
# publish interface object to Ssp core
#
Ssp.routerInit = SillyRouterInit
```

Specifying Router Initialization Scripts on the SAE

To use SDX Configuration Editor to specify router scripts:

1. In the navigation pane, select a directory configuration object for the SAE that you want to configure.
2. Select the Router tab, and expand the Router Scripts section.



3. Edit or accept the default values in the appropriate fields.

   See *Router Script Fields* on page 42.

4. Select **File > Save**.

5. Right-click the configuration file, select **SDX System Configuration > Export to LDAP Directory**.

Router Script Fields

In SDX Configuration Editor, you can edit the following fields in the Router Scripts section of the Router pane in an SAE configuration file.

Extension Path

- Path to router initialization scripts that are not in the default location, */opt/UMC/sae/lib*.
- Value—List of paths separated by semicolons (;)
- Default—No value
- Property name—Extension.path

General Script

- Router initialization script that can be used for all types of routers that the SRC software supports. The script is run when the connection between a router and the SAE is established and again when the connection is dropped.
- Value—Name of a script
- Default—No value
- Property name—Router.script.*

Using SNMP to Retrieve Information from Network Devices

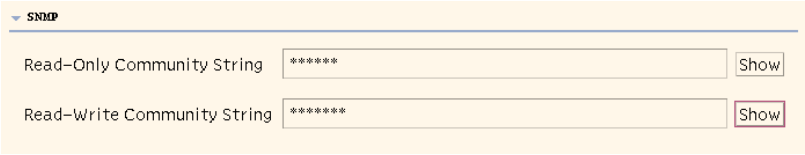
You can use SNMP to retrieve information from a network device. For example, if you create a script that uses SNMP, you need to specify the SNMP communities that are on the network device.

We recommend that you specify SNMP communities for each virtual router object. (See *Adding Virtual Router Objects* on page 30.) You can also configure global default SNMP communities.

Configuring Global SNMP Communities in the SRC Software

You can configure global default SNMP communities that are used if a VR does not exist on the router or the community strings have not been configured for the VR. To use SDX Configuration Editor to configure global default SNMP communities:

1. In the navigation pane, select a directory configuration object for the SAE that you want to configure.
2. Select the Router tab, and expand the SNMP section.



3. Edit or accept the default values in the fields.

   See *Global SNMP Community Fields* on page 43.

4. Select **File > Save**.
5. Right-click the configuration file, and select **SDX System Configuration > Export to LDAP Directory**.

Global SNMP Community Fields

In SDX Configuration Editor, you can edit the following fields in the Router pane in an SAE configuration file.

Read-Only Community String

- Default SNMP community string used for read access to the router.
- Value—SNMP community string that matches a read-only community string configured on the router
- Default—Public
- Property name—Router.read-only.community.string

Read-Write Community String

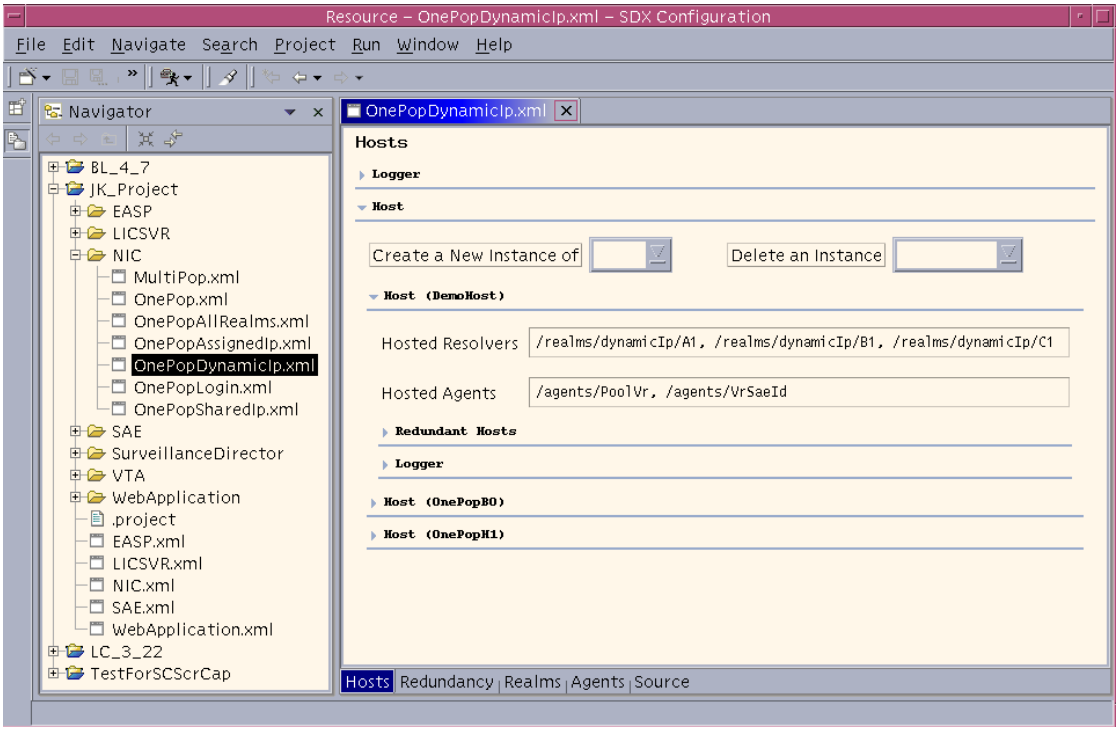
- Default SNMP community string used for write access to the router.
- Value—SNMP community string that matches a read-write community string configured on the router
- Default—Private
- Property name—Router.read-write.community.string

Using the NIC Resolver

If you are using the assigned IP subscriber method of logging in subscribers, and you are using the NIC to determine the subscriber's SAE, you need to configure a resolver on the NIC. The OnePopDynamicIp sample configuration data supports this scenario. The OnePopDynamicIp configuration supports one point of presence (POP) and provides no redundancy. The realm for this configuration accommodates the situation in which IP pools are configured locally on each virtual router object.

You can access the OnePopDynamicIp configuration in either SDX Admin or SDX Configuration Editor. Figure 7 shows the sample configuration in SDX Configuration Editor.

Figure 7: OnePopDynamicIP Sample Configuration in SDX Configuration Editor



For more information about the resolution process for OnePopDynamicIp, see *SDX Network Guide: SAE, Juniper Networks Routers, and NIC*.

Part 2

Integrating Directories

Chapter 3

Overview of LDAP Integration

A directory that implements the Lightweight Directory Access Protocol (LDAP) is the central repository for data shared between the various components in an SRC environment. You can integrate a number of supported third-party directory servers to provide LDAP support.

This chapter contains the following sections:

- LDAP Overview on page 47
- Supported Directories on page 49
- Directory Security on page 49
- Provisioning the Directory on page 50
- Naming Directory Entries on page 50
- Directory Object Model and Schema on page 51

LDAP Overview

The LDAP model is a standard that specifies directory access to servers that comply with the following RFCs:

- RFC 2255—The LDAP URL Format (December 1997)
- RFC 2254—The String Representation of LDAP Search Filters (December 1997)
- RFC 2253—Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names (December 1997)
- RFC 2252—Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions (December 1997)
- RFC 2251—Lightweight Directory Access Protocol (v3) (December 1997)

LDAP is optimized to support searching for information that meets specified criteria.

An LDAP directory is the central integration point for the systems that interact with the SRC software, such as network devices and RADIUS servers, and serves as a repository for customer information, service information, policies, and SRC configuration information, including licensing material. For information about how a directory can be deployed in an SRC configuration, see *SDX Getting Started Guide, Chapter 26, Planning an SRC Installation on a Solaris Platform*.

Because a directory is a critical component in your SRC environment, you should have a good understanding of your directory server and of LDAP before using the SRC software. See the documentation for your directory server. This chapter provides information specific to directory configuration for the SRC software.

Directory Availability

Directory redundancy increases the level of availability and performance for an SRC deployment. A number of SRC components, such as the SAE, rely on access to the directory to obtain configuration and provisioning information. To maintain continuous access to the directory, an SDX directory client can be configured to use one directory server as the primary directory and to use any number of backup directories. The SRC software works with multiple servers in the following way:

- The first server specified is the primary or preferred directory server; any other servers comprise an ordered list of backup servers.
- If the primary directory server is not available or fails, the SRC software tries each of the backup servers in turn according to the ordered list. It switches directory connections to the first available backup directory.
- If a backup directory fails, the SRC software again tries each of the directory servers in turn, beginning with the primary server and proceeding through the ordered list. It switches directory connections to the first available backup directory.
- If the primary directory recovers or becomes available, the directory connection switches back to the primary directory server.

For sample deployments that use one or more backup directories, see *SDX Getting Started Guide, Chapter 26, Planning an SRC Installation on a Solaris Platform*.

Directory Updates

When the SAE starts, objects such as policy and service definitions are loaded in to the directory. Directory data for some other objects, such as retailer and subscriber definitions, are loaded only when needed.

An SDX directory client runs in a number of components. Changes to data that is loaded by a directory client, but that is not loaded on an as-needed basis, can be updated for affected components. Therefore, you do not need to manually reload the data in the SDX directory client.

Depending on the configuration for an object, a client can detect data changes and make appropriate updates. In some cases, you can disable directory updates.

All SAEs in a configuration share the same data and receive the same updated directory information. As a result any SAE can manage a subscriber or a service. For example, when you create a new service, the service definition is stored in the directory, all SAEs are notified, and all active subscriptions to the service are adjusted to the new definition.

Supported Directories

You can directly integrate supported directory servers by installing the directory software to meet SDX specifications and then running a script provided by an SRC add-on component. The SRC software provides prepackaged integration with the following directory servers:

- DirX directory server—See *Chapter 7, Integrating the DirX Directory Server*.
- eTrust Directory—See *Chapter 4, Integrating eTrust Directory*
- Oracle Internet Directory —See *Chapter 5, Integrating Oracle Internet Directory*.
- Sun ONE Directory Server—See *Chapter 6, Integrating Sun ONE Directory Server*.

For information about which directory servers have been tested with the SRC software, see the *SRC-PE Release Notes*.

Directory Security

You can help to secure data in your directory by configuring:

- Directory Access on page 49
- LDAPS Directory Connections on page 50

Directory Access

Directories specify different levels of access for users to particular information in the directory. Access control lists define access rights for users and clients.

From the SRC software, you can configure appropriate authorization for operators to access the directory and specific SRC components. Service providers can set up a multilayered access control scheme for operators. For instance, a network operator might be able to create configuration entries for network devices, but not for services or subscribers. See *SDX Subscribers and Subscriptions Guide, Chapter 13, Configuring Subscribers and Subscriptions with SDX Admin*.

All clients that have the credentials of an SRC component are granted only the level of access required. For example, RADIUS requires access to read and compare user passwords that are part of the RADIUS profiles, but does not require access to other user passwords. RADIUS also does not require access to modify, create, or delete the entries.

For detailed information about directory access, see *Chapter 10, Access Control Scheme*.

Directories also provide audit control to track user activity. Audit control lets you trace the changes that a user makes to the directory. Because the SRC software can support directory access for a number of users, you can use a directory audit control mechanism to determine the actions that a user takes on SDX data, such as modifying directory entries.

LDAPS Directory Connections

LDAPS is LDAP that uses Secure Sockets Layer (SSL) to secure communications between an LDAP client and server. Most directories, including DirX directory server, eTrust Directory, Oracle Internet Server, and Sun ONE Directory Server support LDAP through SSL.

The SAE supports LDAPS connections to the directory server for components within the SAE. The SAE can provide simultaneous LDAP and LDAPS connections for different components. LDAPS connections are useful for protecting confidential data, such as attributes that contain passwords and keys. For public data that does not require the security of SSL, you can configure LDAP rather than LDAPS.

For information about configuring LDAPS connections, see *Chapter 8, Configuring LDAPS for SRC Components*.

Provisioning the Directory

You can provision the directory by:

- Using SRC applications such as SDX Admin, SDX Configuration Editor, and Policy Editor.
- Importing SDX directory files that are in LDAP Data Interchange Format (LDIF) into other programs.

  If you plan to import data into the SDX directory, you should have a good working knowledge of the SDX schema. See the LDAP schema documentation in the SRC software distribution in the folder */SDK/doc/ldap/* or on the Juniper Networks Web site at

  http://www.juniper.net/techpubs/software/management/sdx
- Using an external operations support system (OSS) to provision all or part of the directory information directly through the LDAP interface. Both mechanisms must follow the SDX LDAP schema.
- Integrating data from another storage medium. See *Chapter 9, Integrating Data with the LDAP Directory*.

Naming Directory Entries

When you add an entry to the directory, an asterisk (*) in the name can create more than one match and result in none of the associated configurations being used. Also, the directory does not distinguish between upper case and lower case characters in object names.

When you add an entry to the directory, do not use an asterisk (*) or other non-alphanumeric characters in the name. Do not specify object names that are the same but differ only in case use in the name. For example, do not use myrouter and MyRouter. Java applications and the enterprise service portals do not handle dots (.) and slashes (/) in subscriber names. When you enter the name of a subscriber, including a subscriber folder or a retailer name for a subscriber, do not use a dot or slash in the subscriber name.

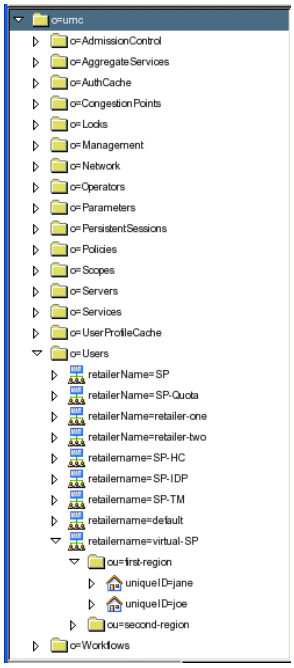
Directory Object Model and Schema

The SDX directory schema is based on X.500/LDAP standards and the Common Information Model version 2.5 (CIM 2.5) schemas. The CIM provides definition of management information for systems, networks, applications, and services. The SDX schema extends the CIM to provide elements for modeling services; residential, enterprise, and retail customers; policies; network elements; and others.

Directory Object Model

The directory object model represents the way objects are stored in a directory. An object comprises data that is stored as entries and organized into a hierarchical structure called a directory information tree (DIT). A DIT contains a number of other trees called subtrees. Figure 8 shows the top-level objects, as well as some subordinate objects in the Users folder in the SDX directory tree as it appears in SDX Admin.

Figure 8: Directory Tree as Displayed by SDX Admin



Each entry has a number of attributes—special characteristics that provide information about the entry. An attribute can also be referred to as a property, as they are in SDX Configuration Editor.

Each entry has an attribute to specify the name for the entry. A name for an entry must be unique within a specified level in the tree hierarchy; for example, each retailer name must be unique with the Users folder, as illustrated in Figure 8.

Naming Convention for Entries

The name for an entry can be expressed as either a relative distinguished name (RDN) or a distinguished name (DN). The RDN identifies a unique entry at one level in the directory tree. Each RDN identifies an attribute type with the associated value. The following list shows sample RDNs from Figure 8:

o = umc
o = Users
retailername = virtual-SP
ou = first-region
unique-id = joe

NOTE: Do not use the "#" character in DNs. It can cause various problems.

A DN is a comma-separated sequence of hierarchical entry names in the tree, concatenated from a specified entry backward to the base, or root, of the tree structure. In contrast to the RDN, the DN for an entry is unique within the entire directory. Each entry in the directory is identified and can be located by its distinguished name (DN). The DN for subscriber Joe would be the following:

unique-id = joe, ou = first-region, retailername = virtual-SP, o = Users, o = umc

NOTE: Throughout the SRC documentation, in text we show the elements of a DN separated by comma/space pairs. We do this for readability. The SRC software and the LDAP specifications require acceptance of the space, but the space is not necessary.

A base DN is the DN of an object that serves as the starting point for a directory search. For the directory as a whole, the base DN is *o = umc* for a default installation of the SRC software; it is the root object of the tree. For a search of policies, the base DN is the following:

o = policies, o = umc

A bind DN is the DN of a login to the directory. This DN must be entered (like a username) with a password to log in to the directory. In the SRC software, you use the bind DN and a password when you access the directory; for example, when you start SDX Admin to view or modify the contents of the directory.

Table 6 lists some of the common DN attribute types.

Table 6: Common DN Attribute Types

Attribute Type Abbreviation	Attribute Type Definition
cn	Common name
o	Organization name
ou	Organizational unit name (In the directory for SDX the organizational unit is typically a directory.)
uid	User ID

Directory Schema

An LDAP schema defines the content and structure of the directory tree. It determines the types of entries that can exist in the directory, the organization for entries, and the relationships between the different types of entries. The directory schema for the SRC software includes entries primarily for management configuration, network device configuration, policies, services, retailers or providers, and subscriber profiles.

Object Classes

Object classes define the different types of entries that can exist in the directory; each entry in the directory belongs to one or more object classes. An object class contains specified attributes to define the characteristics of the entry. For information about attributes, see *Attributes* on page 57.

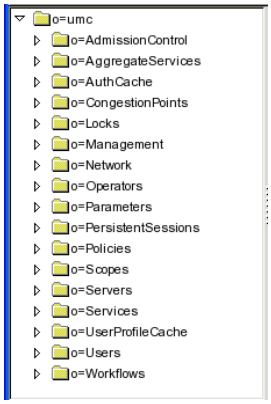
The following sections provide information about the objects in the SDX directory:

- Objects Representing Folders on page 53
- Subscriber Objects on page 54
- Service Objects on page 55
- Subscription Profile Objects on page 55
- Policy Objects on page 55
- Network Device Objects on page 55
- Configuration and System Management on page 56
- Workflow and OSM Schema Elements on page 56

Objects Representing Folders

Folders divide the DIT into logical subtrees. The content prefix of the SDX tree is *o = umc*. The tree is divided into subtrees, such as those for subscribers and service profiles, services, networks, and policies. Figure 9 shows the first-level folders under *o = umc* that are created during the setup of the directory. You can create the second-level folders by using SDX Admin.

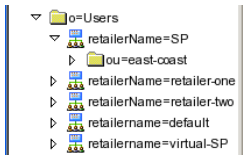
Figure 9: First-Level Folders



The standard object classes organization and organizationalUnit divide trees into subtrees. In some cases, these objects classes do not provide all the attributes required by an entry. An auxiliary object class (an object class that supplies additional information to augment structural object classes) supplies the additional attributes. This moreInformationAuxClass can be attached to the organization object class and the organizationalUnit object class.

You use the object class organizationalUnit to create folders under first-level folders. For example, in Figure 10, the *ou = east-coast* folder is an organizationalUnit object.

Figure 10: Sample Retailer Folders



The DN for this folder is:

ou = east-coast, retailerName = SP, o = Users, o = umc

Subscriber Objects

The directory provides object classes for the categories of subscribers supported by the SRC software, such as:

- Residential users—umcUser
- Enterprises—umcEnterprise
- Sites—umcSite
- Retailers—umcRetailer
- Routers—umcRouterSubscriber

Folders under a umcRetailer object provide a convenient way to organize groups of subscribers. These folders can use the umcSubscriber auxiliary object class.

Subscriber objects are stored under *o = Users, o = umc*.

Service Objects

The umcService object class models is the base class in the service hierarchy. At a high level, SRC provides object classes for services.

Services are also referred to as SSP services (for residential and enterprise users)—sspService.

Service objects are stored under *o = Services, o = umc*. Services can also be stored under *l = < locality >, o = Scopes, o = umc*.

The SRC software supports parameter substitution for services. Parameter substitution requires the attachment of the auxiliary object class parameterAuxClass to an sspService object and to a locality if the sspService object is configured within a scope.

Subscription Profile Objects

When a subscriber subscribes to a service, the SDX subscription component creates an object for the subscription profile from the object class umcServiceProfile. The umcServiceProfile subscription object is created as a subordinate (child) of the subscriber object in the tree. The SRC software supports parameter substitution for service subscriptions, which means that the auxiliary object class parameterAuxClass can be attached to an instance of sspServiceProfile.

Subscriptions are stored under entries subordinate to *o = Users, o = umc*.

Policy Objects

The policy information model for SRC software is based on the Policy Core Information Model (PCIM) that is mapped to the Policy Framework LDAP core schema by the IETF. The SRC software extends this model to produce a policy model that is very close to the one that routers and other network devices use. A policy group object consists of one or more policy lists, which contain one or more policy rules. A policy rule consists of policy conditions and policy actions. The objects policy group, policy list, and policy rules are mapped to structural object classes. Each of these classes is derived from the object class policy.

Policy objects are stored under *o = Policies, o = umc*.

We recommend that you define policies in the directory before you create service objects. Service objects reference policies. The service definition interface should provide LDAP search functionality that retrieves all available policies.

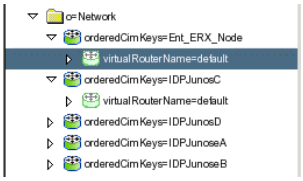
Network Device Objects

Physical network elements are modeled with the CIM Chassis object classes. The object class dlm1Chassis is used for any Juniper Networks equipment, SDX devices, and network devices in the connection path.

Some SRC components, such as the SAE, require additional information about the router, such as virtual routers or interface classifications. The SDX object class umcVirtualRouter is a structural object class that represents virtual routers on Juniper Networks routing platforms. The umcClassificationProfile is an auxiliary object class that is attached to the structural object class.

Network devices are stored in the folder *o = network, o = umc*. Virtual routers are stored subordinate to network devices. Figure 11 shows a sample network folder that contains a number of network devices.

Figure 11: Sample Network Folder



Congestion points in a connection path are modeled by the object class networkInterface, which is subordinate to network device objects in the directory tree. The network devices used for grouping the congestion points are stored in the folder *o = AdmissionControl, o = umc*.

Configuration and System Management

Some of the SDX configuration information, such as license configuration data, is stored in the directory.

The CIM object class dlm1UnitaryComputerSystem represents the hosts on which SAE and system management components, such as an SNMP server, are installed. For management components, the CIM object class dlm1UnitaryComputerSystem replaces the object class umcHost.

The dlm1UnitaryComputerSystem object class stores an IP address in the CIM attribute dlmIdentifyingDescriptions. The location (for example, POP A) is specified in dlmOtherIdentifyingInfo.

All configuration and management objects are stored under *o = Management, o = umc*.

Workflow and OSM Schema Elements

Workflows can be stored in the SDX directory in byte-code format. The umcWorkflow object is a structural object class that represents workflow objects. Workflows and state machines are stored under *o = Workflows, o = umc*. Whenever objects are locked by the object state manager, the locked objects are stored in *o = Locks, o = umc*.

Attributes

An attribute contains a characteristic and the values for that characteristic. Attributes for an object class can be required or optional. An attribute type provides the syntax, sorting, and comparison rules to be used for an attribute.

The following example shows the characteristics and values for the sspType attribute:

- Description—Specifies the provider type (content provider or Internet service provider or others).
- Object identifier—1.3.12.2.1107.1.3.101.10.4.35
- Attribute syntax—Directory String
- Equality matching rule—Case Ignore Match
- Multivalued—False

Structure Rules

DIT structure rules are rules specific to an object class; they specify the location of the object class in the DIT and a name form which identifies naming attributes for an object class. Structure rules state which object classes can be located superior (as a parent) and subordinate (as a child) to other object classes. For example, service profiles are subordinate to users. The DIT structure rules prevent the addition of entries that belong to an object class in an unsupported location in the DIT. If you try to add an entry in an unsupported location, you receive an error message.

The structure rules are used to model dependencies in the DIT. For example, structure rule (SR) 1—organizationalNameForm—allows the creation of the root directory *o = umc*.

Content Rules

A DIT content rule defines which attributes an entry for a specified object class can contain, such as:

- Mandatory attributes that an entry must contain
- Optional attributes that an entry can contain
- Auxiliary object classes that can be associated with the object class
- Optional attributes from the structural and auxiliary object class definitions that an entry must not contain

Where to Find More Information About the Object Model and Directory Schema

The SRC software provides detailed documentation of the object model and the directory schema, including graphical representations of the schema models. See the LDAP schema documentation in the SRC software distribution in the folder */SDK/doc/ldap/* or on the Juniper Networks Web site at

http://www.juniper.net/techpubs/software/management/sdx

The documentation for the SDX object model and schema provides the following topics:

- Attribute types—Lists the attributes types and the associated object classes that appear in the directory.
- Object classes—Provides a link to the directory that contains HTML files that describe each object class. You can also view a list of object classes from the Attribute Types page.
- Content rules—Lists and describes the content rules for structural object classes and the associated auxiliary classes.
- Structure rules—Lists and describes the structure rules for the schema and provides examples of how structure rules are used.
- Name forms—Provides a link to the directory that contains HTML files that describe each name form. You can also view a list of name forms from the structure rules page.
- Schema models—Provides links to graphical representations of the schema models for network, operator, policy, services, user, and workflow in GIF and PDF formats.

The SRC software also provides sample data files in LDIF, an easy-to-read format. The location of the LDIF files on your system depends on the directory integrated with the SRC software. Table 7 lists the location of the LDIF files for the various directories.

Table 7: Location of LDIF Files

Directory Server	Directory That Contains LDIF Files
DirX directory server	<*DIRX_HOME*>*/customize/data* where <DIRX_HOME> is the DirX home directory
eTrust Directory	*/opt/UMC/conf/etrust/sdx_ldif*
Oracle Internet Directory	*/opt/UMC/conf/OID*
SUN ONE Directory Server	*/opt/UMC/conf/iDS*

For detailed information about LDAP, see the documentation for your LDAP server.

Chapter 4

Integrating eTrust Directory

Computer Associates International, Inc provides eTrust Directory, a directory server. Use the information in this chapter to integrate eTrust Directory with Juniper Networks routers and the SRC software. See the *SRC-PE Release Notes* for information about the compatibility of this SRC release with versions of eTrust Directory.

This chapter contains the following sections:

- Overview of Integration with eTrust Directory on page 59
- About the eTrust Directory Add-On Package on page 59
- Integrating the eTrust Directory with the SRC Software on page 61
- Starting SDX eTrust Directory on page 62
- Stopping SDX eTrust Directory on page 63
- Displaying the Status of SDX eTrust Directory on page 63
- Backing Up and Restoring eTrust Directory on page 63

Overview of Integration with eTrust Directory

You can integrate eTrust Directory into your SRC environment by installing the directory software; and then running scripts that are installed from the SRC add-on package for eTrust Directory. eTrust Directory supports LDAP v3 and is designed to provide performance, scalability, and security.

About the eTrust Directory Add-On Package

The eTrust Directory add-on package for the SRC software is named UMCedsa. This package includes two scripts:

- **setup.sh**—Creates and configures SDX eTrust Directory Server Agent (jnprsdx).
- **load**—Customizes SDX eTrust Directory Server Agent by loading files in LDAP Data Interchange Format (LDIF). The */opt/UMC/conf/etrust/sdx_ldif* directory contains these files.

The **setup.sh** script performs the following tasks:

- Creates the initialization file for SDX eTrust Directory Agent in the following directory:

  <eTrust_home_directory>*/dxserver/config/servers*

- Specifies that SDX eTrust Directory Agent use the following schema configuration files located in the file <eTrust_home_directory>*/dxserver/config/schema/jnprsdx.dxg*. This file defines the groups of schema definition files that the SRC software requires:

  - *x500.dxc*—Defined and required by eTrust.
  - *cosine.dxc*—Defined and required by eTrust.
  - *umich.dxc*—Defined and required by eTrust.
  - *inetop.dxc*—Defined and required by eTrust.
  - *dxserver.dxc*—Defined and required by eTrust.
  - *jnprsdx_schema.dxc*—Defined and required by the SRC software. This file contains all SDX attribute and object class definitions.

- Creates the following configuration files and defines configuration settings for SDX eTrust Directory Agent in each file.

  - <eTrust_home_directory>*/dxserver/config/access/jnprsdx.dxc*—Specifies access control parameters.
  - <eTrust_home_directory>*/dxserver/config/database/jnprsdx_index.dxc*—Specifies index configuration.
  - <eTrust_home_directory>*/dxserver/config/database/jnprsdx.dxc*—Specifies jnprsdx as the database name.
  - <eTrust_home_directory>*/dxserver/config/knowledge/jnprsdx.dxc*—Specifies information about connectivity, protocols used, authentication, and the root entry for the directory.
  - <eTrust_home_directory>*/dxserver/config/limits/jnprsdx.dxc*— Specifies operational parameters for the database.
  - <eTrust_home_directory>*/dxserver/config/logging/jnprsdx.dxc*— Specifies logging parameters.
  - <eTrust_home_directory>*/dxserver/config/settings/jnprsdx.dxc*— Specifies default settings for the database. These entries include parameters specific to the SRC software.

- Specifies operating parameters that include size and time limits and default settings for SDX eTrust Directory Agent.

NOTE: You configure replication within eTrust Directory. See the documentation for eTrust Directory.

Integrating the eTrust Directory with the SRC Software

The tasks to integrate the eTrust Directory software with the SRC software are:

1. Installing eTrust Directory to Integrate with the SRC Software on page 61
2. Configuring the SDX eTrust Directory Server Agent with the SRC Software on page 61

Installing eTrust Directory to Integrate with the SRC Software

To install eTrust Directory:

- Use the eTrust express installation to install eTrust Directory on your Solaris platform. See the documentation for eTrust Directory.

  The installation process creates a user ID named dsa that is the administrator user ID for eTrust Directory. Assign a password to this user ID.

Configuring the SDX eTrust Directory Server Agent with the SRC Software

You integrate eTrust Directory with the SRC software by running commands provided by the eTrust Directory add-on package.

Before you run these commands, ensure that the add-on package is installed on the system on which the eTrust Directory is installed. For information about installing the package, see *SDX Getting Started Guide, Chapter 28, Installing the SRC Software on a Solaris Platform*.

To integrate eTrust Directory:

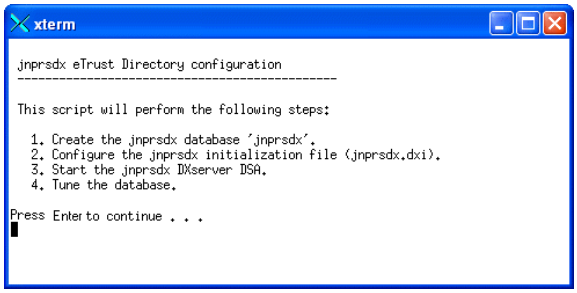
1. Navigate to the following directory:

   /opt/UMC/conf/etrust

2. Log in using the **dsa** user ID and the associated password, and run the **setup.sh** script by entering the following command:

   ./setup.sh

When the script starts, you see the following information on the screen:



3. Login as root and run the **load** script by entering the following command:

   ./load

4. Select sample data to load when prompted.

   When the sample data is loading, you may see messages that indicate that some entries are already present. The **load** script generates these messages for different LDIF files that contain the same data. You can ignore these messages.

For information about the tasks that the script completes, see *About the eTrust Directory Add-On Package* on page 59.

Starting SDX eTrust Directory

To start SDX eTrust Directory:

1. Navigate to the SDX eTrust installation directory (*/opt/UMC/conf/etrust* by default).

2. Log in using the **dsa** user ID and the associated password.

3. Start the SDX eTrust Directory Agent from its installation directory.

   /opt/UMC/conf/etrust/ldap start

For more information about starting an instance of eTrust Directory, see the product documentation for the directory.

Stopping SDX eTrust Directory

To stop SDX eTrust Directory:

1. Navigate to the SDX eTrust installation directory (*/opt/UMC/conf/etrust* by default).
2. Log in using the **dsa** user ID and the associated password.
3. Stop the SDX eTrust Directory Agent from its installation directory.

   /opt/UMC/conf/etrust/ldap stop

For more information about stopping an instance of eTrust Directory, see the product documentation for the directory.

Displaying the Status of SDX eTrust Directory

To display the status of SDX eTrust Directory:

1. Navigate to the SDX eTrust installation directory (*/opt/UMC/conf/etrust* by default).
2. Log in using the **dsa** user ID and the associated password.
3. Display the status of the SDX eTrust Directory Agent from its installation directory.

   /opt/UMC/conf/etrust/ldap status

Backing Up and Restoring eTrust Directory

For information about how to backup and restore an eTrust Directory, see the product documentation for the directory.

We recommend that you restart SRC components after restoring directory data from a backup, to ensure that the restored database is used.

Chapter 5

Integrating Oracle Internet Directory

Oracle Internet Directory is a software component in the Oracle Application Server 10g by Oracle Corporation. Use the information in this chapter to integrate Oracle Internet Directory with Juniper Networks routers and the SRC software. See the *SRC-PE Release Notes* for information about the compatibility of this SRC release with versions of Oracle Internet Directory.

This chapter contains the following sections:

- Overview of Oracle Internet Directory Integration on page 65
- About the Oracle Internet Directory Add-On Package on page 66
- Integrating the Oracle Internet Directory with the SRC Software on page 66
- Starting and Stopping Oracle Internet Directory on page 68
- Setting Up Local Configuration for SRC Components on page 68
- Backing Up and Restoring the Oracle Internet Directory on page 69

Overview of Oracle Internet Directory Integration

You can integrate Oracle Internet Directory into your SRC environment by installing the directory software and selecting configuration values to support integration with the SRC software. After you install the Oracle Internet Directory, you run a **load** script that is installed from the SRC add-on package for Oracle Internet Directory. Oracle Internet Directory supports LDAP v3 and provides scalability, reliability, and flexibility.

If you use a RADIUS server and use Oracle Internet Directory in your environment, you can use a supported RADIUS server other than Merit AAA Server. Merit AAA Server requires that passwords be stored as clear text; however, the installation for Oracle Internet Directory recommended for integration with the SRC software uses password encryption.

About the Oracle Internet Directory Add-On Package

The Oracle Internet Directory add-on package for the SRC software is named UMCOIDa. This package provides a **load** script and files in LDAP Data Interchange Format (LDIF) to integrate Oracle Internet Directory with the SRC software. The */opt/UMC/conf/OID* directory contains these files. If you try to run the **load** script for Oracle Internet Directory and the directory software is not installed, you receive a message to indicate that the directory is not installed on your host.

The **load** script performs the following tasks:

- Defines access permission to Oracle Internet Directory by using the following values:
  - Root DN—cn = oracladmin
  - Root password—admin123
  - LDAP port number—389

  
  You configure access controls within Oracle Internet Directory. See the documentation for Oracle Internet Directory.
- Sets password encryption to Secure Hash Algorithm (SHA)
- Disables checking of the password syntax to make syntax compatible with the length of passwords in the sample data
- Extends the LDAP schema to:
  - Add classes and attribute types
  - Index SDX attributes
  - Create the directory infrastructure for SDX entries
- Lets you load sample data

Integrating the Oracle Internet Directory with the SRC Software

The tasks to integrate the Oracle Internet Directory software with the SRC software are:

1. Installing Oracle Internet Directory to Integrate with the SRC Software on page 67
2. Running the Load Script for the Oracle Internet Directory Integration on page 68

Installing Oracle Internet Directory to Integrate with the SRC Software

For information about the Oracle Internet Directory component of Oracle Application Server 10g, see the Oracle Web site at:

http://www.oracle.com/appserver/index.html

Before You Install Oracle Internet Directory

Before you install Oracle Internet Directory, complete the following tasks from Solaris Management Console. You can start Solaris Management Console by using the command **usr/sadm/bin/smc**.

1. Create a new group **oinstall** to identify users who have privileges to install Oracle Internet Directory.

2. Create a new user named **oracle** with the password **admin123** who belongs to the oinstall group.

Specifying Configuration Values During Installation

To install and configure Oracle Internet Directory, see the documentation for the product. During the installation, enter values in the installation program for the configuration fields described in the following list. Complete values for other fields as required for your environment.

- Components to install—Component group that includes the Internet Directory and the metadata repository; for example, Identity Management and Metadata Repository.

- NameSpace in the Oracle Internet Directory—UMC

- Database naming values for the global database name and the Oracle System Identifier (SID)—Values that comply with a domain name. For example, for a Juniper domain you could specify a global database name of sdxdb.juniper, and an SID of sdxdb.

- Database character set—Unicode (UTF-8)

- Database schema passwords for SYS, SYSTEM, SYSMAN, and DBSNMP—admin123

- Instance name and password:

  - Instance name—sdx

  - ias_admin password—admin123

If you are setting up a redundant configuration, see the documentation for Oracle Internet Directory.

Verifying Directory Settings

After you install Oracle Internet Directory, the LDAP server and the database instance are running. Verify the configuration values for port that the directory uses, the superadministrator DN, and the password.

To verify configuration information:

1. Open the *<ORACLE_HOME>/OraHome1/install/portlist.ini* file.
2. Verify that the following values are set in the file:
   - Oracle Internet Directory port—389
   - DN for superadministrator—cn = orcladmin
   - Password for superadministrator—admin123

Running the Load Script for the Oracle Internet Directory Integration

You integrate Oracle Internet Directory with the SRC software by running the **load** command for the Oracle Internet Directory add-on package. Before you run this command the add-on package must be installed on the system on which Oracle Internet Directory runs. For information about installing the package, see *SDX Getting Started Guide, Chapter 28, Installing the SRC Software on a Solaris Platform*.

To run the **load** script:

1. Enter the following command:

   /opt/UMC/conf/OID/load

   If the **load** script does not encounter any errors, you receive a message that indicates that the script set the Oracle Internet Directory home directory. This message also lists the DN for superadministrator, password for superadministrator, and Oracle Internet Directory port. Otherwise, messages notify you of errors detected.

2. Select sample data to load when prompted.

For information about the tasks that the script completes, see *About the Oracle Internet Directory Add-On Package* on page 66.

Starting and Stopping Oracle Internet Directory

For information about how to start and stop an instance of Oracle Internet Directory, see the documentation for that product.

Setting Up Local Configuration for SRC Components

When you set up the local configuration for SRC components, such as license server, the SAE, and NIC, as well as RADIUS and Web applications, you specify information for the directory configuration. For components to use Oracle Internet Directory, you specify the configuration directory password as admin123.

Backing Up and Restoring the Oracle Internet Directory

For information about how to backup and restore an Oracle Internet Directory, see the product documentation for the directory. Refer to the Oracle Web site at:

http://www.oracle.com/appserver/index.html

We recommend that you restart SRC components after restoring directory data from a backup, to ensure that the restored database is used.

Chapter 6

Integrating Sun ONE Directory Server

Sun ONE Directory Server is a software product by Sun Microsystems, Inc. that provides a central repository for storing and managing identity profiles, access privileges, and application and network resource information. This chapter describes how to integrate Sun ONE Directory Server with Juniper Networks routers and the SRC software.

For information about compatibility of this SRC release with Sun ONE Directory Server releases, see the *SRC-PE Release Notes*.

Topics in this chapter include:

- Overview of Sun ONE Directory Server Integration on page 71
- Integrating the Sun ONE Directory with the SRC Software on page 73
- Starting Sun ONE Directory Server on page 75
- Stopping Sun ONE Directory Server on page 76
- Restarting Sun ONE Directory Server on page 76
- Backing Up the Sun ONE Database on page 76
- Restoring the Sun ONE Database on page 77

Overview of Sun ONE Directory Server Integration

You can integrate the Sun ONE Directory Server product into your SRC environment by installing an SRC add-on package and then installing Sun ONE Directory Server as specified in this chapter. Sun ONE Directory Server is based on industry-standard LDAP and provides advanced security features, carrier-grade scalability, performance, and availability. Sun ONE acts as a central repository for the consolidation of subscriber profiles.

You can use the information stored in Sun ONE Directory Server for the authentication and authorization of subscribers to enable secure access to enterprise and Internet services. Sun ONE helps to ensure that appropriate access control policies are enforced across all communities, applications, and services on a global basis.

About the Sun ONE Add-On Package

The Sun ONE Directory Server add-on package for the SRC software is called UMCiDSa. This package provides integration files for Sun ONE Directory Server versions 5.1 and 5.2:

- An *sdx.inf* file, which integrates with Sun ONE's silent installation feature.
  - For Sun ONE Directory Server versions 5.1—*/opt/UMC/conf/iDS/SunOne5.1/sdx.inf*
  - Sun ONE Directory Server versions 5.2—*/opt/UMC/conf/iDS/SunOne5.2/sdx.inf*
- A load script and files in LDAP Data Interchange Format (LDIF) to integrate Sun ONE Directory Server with the SRC software in the */opt/UMC/conf/iDS* directory.

Silent Installation for Sun ONE Directory Server

Sun ONE's silent installation feature allows Sun ONE software to be embedded with the SRC software through an *sdx.inf* file specific to the version of Sun ONE DIrectory Server that is being installed. No user intervention is required during the setup process. Table 8 describes important setup script information provided in an *sdx.inf* file.

Table 8: Information Provided for the Sun ONE Setup Script

Configuration Property	Value
Installation path	*/opt/UMC/iDS*
Directory configuration administrator (Sun ONE entity)	admin
Password for directory configuration administrator	admin
LDAP port to be used for directory instance	389
Server identifier (Sun ONE specific). The directory instance is installed in the path: */opt/UMC/iDS/slapd-sdx*	sdx
Suffix for new LDAP directory instance	*o = umc*
Identifier for superadministrator	*cn = umcAdmin, o = umc*
Password for superadministrator	admin123
Administrator's port	6666

NOTE: The uid-uniqueness plug-in is not enabled within the initial configuration of the Sun ONE Directory Server software. Because the SRC software does not require a globally unique user ID, this feature should remain disabled.

Load Script to Integrate Sun ONE Directory Server

The **load** script performs the following tasks:

- Configures the password storage mechanism not to use encryption.

  Because the Merit AAA Server (RADIUS) requires that passwords be stored as clear text, the **load** script changes a setting to not store the password in an encrypted manner.

- Extends the LDAP schema to:
  - Add SDX schema requirements
  - Index SDX attributes
  - Create the directory infrastructure for SDX entries
  - Load access control information

    Sun One Directory Server stores the access control information in the aci attribute, which is available for all directory entries. The load script processes the *access.ldif file* to add required access control information.

- Lets you load sample data

The **load** script is designed to work with Sun ONE Directory Server 5.1 and 5.2. For version 5.1, you enter the command with the **5.1** option:

```
load 5.1
```

For version 5.2, you enter the command without any options:

```
load
```

Integrating the Sun ONE Directory with the SRC Software

The tasks to integrate the Sun ONE Directory Server with the SRC software are:

1. Installing the Solaris operating system and the patches that the Sun ONE directory server requires. See the documentation for Sun ONE directory Server for details.
2. Installing the Sun ONE Directory Add-On Package on page 74
3. Configuring an Instance of Sun ONE Directory Server on page 74

Before you can integrate the Sun ONE Directory Server software with the SRC software, you must have access to the Sun ONE Directory Server software; the Sun ONE Directory Server software is not included with your SRC software. See the *SRC-PE Release Notes* for information about the versions of Sun ONE Directory Server and the associated service packs supported.

Obtain the software by downloading the Sun ONE Directory Server software from the Sun Microsystems Web site at:

http://www.sun.com/download

Installing the Sun ONE Directory Add-On Package

You must install the Sun ONE Directory add-on package for the SRC software before you install Sun One Directory Server. For information about installing the package, see *SDX Getting Started Guide, Chapter 28, Installing the SRC Software on a Solaris Platform*.

Configuring an Instance of Sun ONE Directory Server

You must create a new instance of the Sun ONE Directory to integrate the Sun ONE Directory Server software with the SRC software.

To create an instance of Sun ONE Directory and integrate it with the SRC software:

1. Uncompress the archive file that you downloaded from the Sun Microsystems Web site by executing the command:

   gzip -dc <filename>.tar.gz | tar xvf -

   where <filename> is the name of the TAR file.

2. Move to the directory that contains the expanded files.

   For example, if you saved the downloaded file into the directory */tmp/DS*, enter:

   cd /tmp/DS

3. Enter the command appropriate to the version of Sun ONE Directory Server to install an instance of the directory by using an *sdx.inf* file.

   - For Sun One Directory Server 5.1:

     ./setup –s –f /opt/UMC/conf/SunOne5.1/sdx.inf

   - For Sun One Directory Server 5.2:

     ./setup –nodisplay –noconsole –state /opt/UMC/conf/SunOne5.2/sdx.inf

4. Move to the following directory:

   /opt/UMC/conf/iDS

5. Enter the command appropriate to the version of Sun ONE Directory Server to run a **load** script.

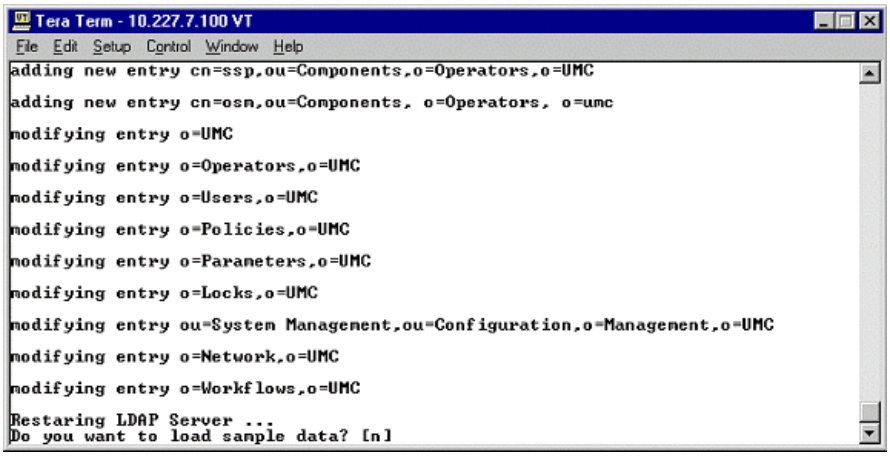
   - For Sun One Directory Server 5.1:

     ./load 5.1

   - For Sun One Directory Server 5.2:

     ./load

After the script updates the schema, loads the directory infrastructure, and creates the access controls, the script prompts you to load the sample data. An example is shown in the following window.



After the script finishes running, SRC components can use the Sun ONE Directory Server.

Starting Sun ONE Directory Server

To start Sun ONE Directory Server:

1. On the Sun ONE Directory Server host, log in as `root` or as an authorized nonroot admin user.

2. Start the Sun ONE Directory Server from its installation directory.

   /opt/UMC/conf/iDS/ldap start

Stopping Sun ONE Directory Server

To stop Sun ONE Directory Server:

1. On the Sun ONE Directory Server host, log in as `root` or as an authorized nonroot admin user.
2. Stop the Sun ONE Directory Server from its installation directory.

   /opt/UMC/conf/iDS/ldap stop

Restarting Sun ONE Directory Server

To restart Sun ONE Directory Server:

1. On the Sun ONE Directory Server host, log in as `root` or as an authorized nonroot admin user.
2. Restart the Sun ONE Directory Server from its installation directory.

   /opt/UMC/conf/iDS/ldap restart

Backing Up the Sun ONE Database

You can manually back up the database for any directory you have installed. For information about migrating a directory database to another host, see *SDX Getting Started Guide, Chapter 35, Upgrading the SRC Software on a Solaris Platform*.

To back up the Sun ONE (formerly iPlanet) database:

1. Log in as `root`.
2. Access the database folder.

   cd /opt/UMC/iDS/slapd-sdx

3. Back up the database.

   ./db2bak

   This script makes a copy of the database and stores it in the following location:

   /opt/UMC/iDS/slapd-sdx/bak/YYYY_MM_DD_HHMMSS

   The backup directory identifies the date (YYYY_MM_DD) and time (HHMMSS) when the backup was created.

Restoring the Sun ONE Database

You can manually restore the database for any directory you have installed. For information about migrating a directory database to another host, see *SDX Getting Started Guide, Chapter 35, Upgrading the SRC Software on a Solaris Platform*.

To restore the Sun ONE database:

1. Log in as `root`.
2. Access the database folder.

   cd /opt/UMC/iDS/slapd-sdx

3. Verify that the Sun ONE Directory Server is shut down. If it is not, shut it down.

   ./stop-slapd

4. Make sure you know the exact backup directory.
5. Run the *bak2db* script by typing:

   ./bak2db /opt/UMC/iDS/slapd-sdx/bak/YYYY_MM_DD_HHMMSS

   where YYYY_MM_DD_HHMMSS identify the date and time when the database backup was created.

6. Start the Sun ONE server.

   ./start-slapd

We recommend that you restart SRC components after restoring directory data from a backup, to ensure that the restored database is used.

Chapter 7

Integrating the DirX Directory Server

DirX Solutions is a product family by Siemens that provides a central repository for storing, managing, and distributing identity profiles, access privileges, and application and network resource information. DirX Solutions contains the software products DirX 6.0 directory server and DirXmetahub 6.0 (the DirX meta engine).

Use the information in this chapter to integrate DirX with Juniper Networks routers and the SRC software. See the *SRC-PE Release Notes* for information about compatibility of this SRC release with DirX releases.

This chapter contains the following sections:

- Overview of DirX Directory Server Integration on page 79
- About the DirX Add-On Package on page 80
- Integrating the DirX Directory with the SRC Software on page 80
- Provisioning the Directory by Using DirXmetahub on page 83
- Uninstalling the DirX Directory Server on page 84
- Starting the DirX Directory Server on page 84
- Stopping the DirX Directory Server on page 85
- Backing Up the DirX Database on page 86
- Restoring the DirX Directory Database on page 86

Overview of DirX Directory Server Integration

You can integrate the DirX directory server into your SRC environment by installing the directory software and selecting configuration values to support integration with the SRC software. After you install the DirX software, you run the generate script that is installed from an SRC add-on package.

The DirX directory server is based on LDAPv3, Directory Services Markup Language version 2 (DSMLv2), and X.500 directory server standards. You can use DirX to set up a distributed, replicated directory service and to manage data in directories. DirX provides an identity management system to store information about people, organizations, applications, network devices, and other distributed services. Working with the DirX directory server, DirXmetahub decreases the time and effort to operate multiple directories simultaneously.

About the DirX Add-On Package

The DirX directory server add-on package for the SRC software is named UMCdirxa.This package provides a **generate.sh** script that integrates DirX with the SRC software and a *variables.tcl* file that contains configuration information that the SRC software requires. Table 9 describes important generate-script information, which is contained in the *variables.tcl* file. The installation for the add-on package places this file in the *customize* in the dirx user home directory */export/home/dirx*.

Table 9: Information Needed for DirX Generate Script

Configuration Property	Value
LDAP port to be used for directory instance	389
Suffix for new LDAP directory instance	*/o = umc*
Identifier for superadministrator	*/o = umc, cn = umcAdmin*
Password for superadministrator	admin123

The **generate.sh** script also extends the LDAP schema to:

- Index SDX attributes by processing the *schema.adm* file.
- Defines the directory tree structure by processing the *initialize.cp* file.
- Adds access control information by processing the *access.cp* file.
- Creates the directory infrastructure.
- Lets you load sample data.

Integrating the DirX Directory with the SRC Software

The tasks to integrate DirX Directory Server with the SRC software are:

1. Preparing to Install the DirX Directory Server on page 81
2. Installing the DirX Directory Server on page 82
3. Installing the UMCdirxa Add-On Package on page 82
4. Configuring the DirX Directory Server on page 82

You can obtain the DirX Solutions software package by contacting your local Siemens sales representative. For immediate information, go to:

http://www.siemens.com/directory

You can also e-mail Siemens at:

directory@icn.siemens.de

Preparing to Install the DirX Directory Server

Only root users can install the DirX directory server software. To prepare for installation:

1. Log in as `root`.
2. Load the Siemens DirX Solutions software CD, and access the CD directory. For example:

   cd /cdrom/cdrom0

3. Create a new directory user. You must create a directory user called *dirx*. Once the software is installed, the dirx user can manage the DirX software.

   To create a new user and prepare the host for the DirX software:

   a. Start the Solaris Admintool:

      admintool &

      The Admintool: Users window appears.

   b. Select Edit, and then click **Add**.

      The Admintool: Add User dialog box appears.

   c. In the User Name field, enter:

      dirx

   d. In the Set Path field, enter:

      /export/home/dirx

   e. Click **OK**.

      The window closes, and the main Admintool: Users window appears with the information you just entered.

   f. Set the password. For example:

      passwd dirx

   g. Enter and confirm the password when prompted by the Admintool.

Installing the DirX Directory Server

The default base directory for installation is */opt/dirx*. We recommend that you install the DirX software in the home directory of the DirX user as described in *Preparing to Install the DirX Directory Server* on page 81, and install the DirX software package in the home directory */export/home/dirx*.

The directory and log files are created in subdirectories of the installation directory. Before you start the installation, verify that you have sufficient disk space available in the installation directory.

To install the DirX directory server software:

1. Log in as `superuser`.
2. Install DirX with the pkgadd tool. For example:

   pkgadd -d /cdrom/cdrom0/server/dirx/sun/dirx60*

3. Respond to the prompts as listed in Table 10.

Table 10: Prompts for Installation of DirX-SV Server Package

Prompt	Response
Where should the package DirX-SV be installed?	1 = home directory 2 = /opt/dirx **NOTE:** Select 1.
Please enter the login name of an existing user.	dirx
Do you want to install these as setuid.setgid files?	y
Do you want to continue with the installation of < DirX-SV > ?	y

Installing the UMCdirxa Add-On Package

You must install the DirX software before you install the SRC add-on package for DirX. To install the add-on package:

1. Log in as `superuser`.
2. Install the DirX add-on package by following the directions in *SDX Getting Started Guide, Chapter 28, Installing the SRC Software on a Solaris Platform*.

The DirX add-on package is installed in a subdirectory named *customize* in the dirx user home directory */export/home/dirx*.

Configuring the DirX Directory Server

To configure the DirX directory server:

1. Log in as `dirx`, and enter the following commands:

   su - dirx
   cd customize
   sh ./generate.sh

2. At the system prompt "Do you want to load sample data? [n]", enter y for yes.

   The script takes a few minutes to create, initialize, and configure the database and start the server processes. The **generate.sh** scripts generate the *log.txt* log file.

   If the generation fails, the system returns a FAILED message.

3. Verify whether the integration was successful by opening *log.txt* file with an editor, such as vi, and search for FAILED.

   If the search does not find the word FAILED, the integration was successful.

4. To verify that the DirX directory server processes are running, enter:

   ps -ef | grep dirx

   The system displays three processes, as shown in the following sample screen output:

   ```
   root 20263 20260 0 Feb 11 ? 5:40 /export/home/dirx/bin/dirxldapv3
   dirx 20261 20260 0 Feb 11 ?26:26 /export/home/dirx/bin/dirxdsa -d
   /export/home/dirx/server/DB
   dirx 20260 143 0 Feb 11 ?0:00 /export/home/dirx/bin/dirxdsas -d
   /export/home/dirx
   ```

   Wait for the output of all three processes to appear. The DirX LDAPv3 process takes longer than the other two.

Provisioning the Directory by Using DirXmetahub

Metahub provides the ability to integrate connected data sources, such as a relational database that holds subscriber information, other LDAP directories, or flat files (for example, XML) into the SDX directory infrastructure. Metahub is a set of components that includes:

- Metadirectory store—SDX directory that holds all the required SDX-related information.
- Metaagent—Interface to the connected data source. Its function is to import and export data from the data source.
- Metacontroller—Scriptable directory that joins the engine that transforms the data representation from the connected data source to the SDX LDAP schema. It performs the load, join, and aggregate function on directory entries and attributes.

DirXmetahub provides the SRC software with a unified view of the data by synchronizing the OSS data that is stored in one or more database into the SDX directory. This synchronization process can be performed in a scheduled manner on all data or only data that has changed since the last synchronization.

Uninstalling the DirX Directory Server

To uninstall DirX:

1. Log in as `root`.
2. Stop the DirX directory server. For example:

   /etc/init.d/dirx stop

3. Start the Solaris software management tool.

   swmtool

   The Admintool: Software window appears.

4. Select the installed DirX directory server packages.
5. Select **Edit**, and click **Delete**.

   A status dialog box prompts you to confirm that you want to delete the packages.

6. Click **Delete**.

Starting the DirX Directory Server

You can start the DirX directory server in two ways: in a dirx user environment or in a superuser environment.

NOTE: See the Siemens DirX directory server documentation for operating details.

Starting the DirX Directory Server in a dirx user Environment

To start DirX within a dirx user environment:

1. Log in as a dirx user:

   login dirx

2. Change the directory to the *customize* directory:

   cd customize

3. Start DirX by entering the command:

   dirxadm -c start

Starting the DirX Directory Server in a Superuser Environment

To start DirX within a superuser environment:

1. Log in as `root`.
2. Start the DirX directory server:

   /etc/init.d/dirx start

   The start process takes approximately 30 seconds.

Stopping the DirX Directory Server

You can stop the DirX directory server in two ways: in a dirx user environment or in a superuser environment.

NOTE: See the Siemens DirX directory server documentation for operating details.

Stopping the DirX Directory Server in a dirx user Environment

To stop DirX within a dirx user environment:

1. Log in as a dirx user:

   login dirx

2. Change the directory to the *customize* directory:

   cd customize

3. Stop DirX by entering the command:

   dirxadm shutdown.tcl

Stopping the DirX Directory Server in a Superuser Environment

To stop DirX within a superuser environment:

1. Log in as `root`.
2. Stop the DirX directory server:

   /etc/init.d/dirx stop

Backing Up the DirX Database

You can manually back up the database for any directory you have installed. For information about migrating a directory database to another host, see *SDX Getting Started Guide, Chapter 35, Upgrading the SRC Software on a Solaris Platform*.

To back up the DirX database:

1. Log in as user `dirx`, and access the *customize* subdirectory.

   cd customize

2. Archive the database.

   dirxadm
   dirxadm> source bind.tcl
   dirxadm> save -file /tmp/dirxdb

Restoring the DirX Directory Database

You can manually restore the database for any directory you have installed. For information about migrating a directory database to another host, see *SDX Getting Started Guide, Chapter 35, Upgrading the SRC Software on a Solaris Platform*.

To restore the DirX database:

1. Verify that the DirX server is running. See your DirX documentation for details.

2. Restore the archive.

   dirxadm
   dirxadm> source bind.tcl
   dirxadm> restore -file /tmp/dirxdb

We recommend that you restart SRC components after restoring directory data from a backup, to ensure that the restored database is used.

Chapter 8

Configuring LDAPS for SRC Components

LDAPS is LDAP that uses secure sockets layer (SSL) to secure communications between an LDAP client and server. You can configure particular SAE components to use LDAPS to connect to the directory.

This chapter contains the following sections:

- Overview of LDAPS Support on page 87
- LDAPS Authentication and Connection on page 87
- Configuring LDAPS Connections on page 88

Overview of LDAPS Support

The SAE supports LDAPS connections to the directory server for its components, and can provide simultaneous LDAP and LDAPS connections for different components. You can configure the SAE to use LDAPS for some directory connections and LDAP for other directory connections. When planning whether to use an LDAP or LDAPS connection, consider that LDAPS connections have higher processing requirements, use more network bandwidth, and are slower than LDAP connections.

LDAPS connections are useful for protecting confidential data such as attributes that contain passwords and keys. For example, if you want data exchanged between a component such as User Data Manager and the directory to be more secure, you can configure the connection to use LDAPS. For public data that does not require the security of SSL (such as a directory connection that transmits only service information), you can configure LDAP rather than LDAPS.

Most directories, including Oracle Internet Directory, Sun ONE Directory Server, and DirX support LDAP connections through SSL.

LDAPS Authentication and Connection

The steps in the LDAPS authentication and connection sequence are:

1. The directory client initiates LDAPS connection.
2. The directory server sends the X.509 SSL server certificate that it has received from a certificate authority (CA).

3. The client checks the certificate against its trust certificate store. If it matches, the certificate is trusted.
4. The client proceeds with establishing the SSL connection.
5. When the SSL connection is up, the client sends a bind DN and password to the server to establish the LDAP connection.
6. The server authenticates the client and establishes the LDAP over SSL connection.

NOTE: The SRC software does not support certificate authentication for directory clients.

Configuring LDAPS Connections

The tasks to configure LDAPS connections are:

1. Configuring the Directory Server to Support LDAPS Connections on page 88
2. Establishing Trust for Directory Clients on page 89
3. Configuring the SAE to Find the Certificate Store on page 89
4. Enabling LDAPS Communication for SAE Components on page 90
5. Disabling LDAPS Communication for SAE Components on page 91

Configuring the Directory Server to Support LDAPS Connections

For the SAE to communicate with a directory over LDAPS, typically you must configure your directory server to support SSL connections by:

- Obtaining a signed certificate for the directory server from a CA.

  There are many well-known CAs. You can also set up your own CA to sign the directory certificate. The CA must be trusted by the directory clients that use LDAPS to communicate with the directory. Tools such as OpenSSL (http://www.openldap.org) are available to set up a CA.

- Setting up the directory server with an X.509 SSL server certificate. Typically, you install a certificate for the server, and configure the directory server to trust the CA's certificate.
- Enabling SSL.

For information about how to perform these tasks, see the documentation for your directory server.

Establishing Trust for Directory Clients

Each directory client must have a certificate database and must trust the CA to use SSL connections to the directory server.

The SAE, like other Java applications, implicitly trusts certificates that are stored in the */jre/lib/security/cacerts* certificate file. This file is a Java Runtime Environment (JRE) systemwide certificate trust store. By default, the file contains certificates from well-known CAs. If a certificate for the CA that you use for the directory server is available in *cacerts*, view the file on the host on which you installed the JRE.

If your CA is not in the *cacerts* file, you can import the CA into this file or into any certificate store that is in Java Keystores (JKS) format (supported by the Java 2 Software Development Kit). All Java applications running in a specified JRE trust all CAs present in the *cacerts* file.

You can also store a CA certificate in a location other than the default *cacerts* file. You might consider storing the CA elsewhere if you want your SAE to trust only the certificate for the CA that signs the directory server's certificate, or if you do not want other applications that are running in the same JRE to trust the CA's certificate.

To import a CA certificate into a store other than the default *cacerts* file:

- Use the Java **keytool** command.

The following example imports the CA's certificate *ca.crt* into a trust store named *ldapclient.keystore*.

```
keytool -import -v -trustcacerts -alias saeldap -noprompt -file ./ca.crt -keystore
ldapclient.keystore -storepass zaqwsx
```

For more information about the **keytool** command, see

http://java.sun.com/j2se/1.4.1/docs/tooldocs/solaris/keytool.html

Configuring the SAE to Find the Certificate Store

To enable the SAE to locate the certificate store, edit the */opt/UMC/sae/etc/default.properties* file.

To use a certificate file other than the default:

- In the */opt/UMC/sae/etc/default.properties* file, specify the name and path of the file in the Security.ssl.trustcertstore property.

  The following example specifies that the SAE use the *trustcerts* file:

  ```
  Security.ssl.trustcertstore = /opt/UMC/sae/etc/trustcerts
  ```

To specify that the SAE use the default *cacerts* file:

- In the */opt/UMC/sae/etc/default.properties* file, add a comment character before the Security.ssl.trustcertstore property.

Enabling LDAPS Communication for SAE Components

To enable an LDAPS connection for an SAE component, you edit the security properties for the component. How you enable the properties depends on the component for which you are enabling LDAPS.

To enable an LDAPS connection for a component:

1. Open the configuration for the security properties for the component.

   Table 11 shows how to access the security properties for the various components.

Table 11: How to Access Security Properties for SRC Components

SRC Component	How to Enable Security Properties
■ Configuration Manager	■ Edit the */opt/UMC/sae/etc/default.properties* file.
■ User Data Manager ■ Equipment Data Manager ■ Service Data Manager ■ LDAP Authentication Plug-in ■ License Manager	1. In SDX Admin, select a configuration object (such as *l = POP_ID*) under *l = SAE, ou = staticConfiguration, ou = Configuration, o = Management, o = umc*. 2. Select the Main tab. The security properties appear in the list of properties on the Main tab.
■ Enterprise Service Portal User Data Manager ■ Enterprise Service Portal Service Data Manager	1. In SDX Admin, select *l = EASP, ou = staticConfiguration, ou = Configuration, o = Management, o = umc*. 2. Select the Main tab. The security properties appear in the list of properties on the Main tab.

2. Remove the comment character (#) that appears before the component's security protocol property. See Table 12.

Table 12: Security Protocol Properties for SAE Components

SAE Component	Security Protocol Property
Configuration Manager	Conf.directory.security.protocol
User Data Manager	UserDataSource.repository.ldap.server.security.protocol
Equipment Data Manager	UserCacheDataSource.repository.ldap.server.security.protocol
Service Data Manager	ServiceDataSource.repository.ldap.server.security.protocol
LDAP Authentication Plug-In	Plugin.ldapAuth.securityProtocol
License Manager	LicenseMgr.repository.ldap.server.security.protocol
Enterprise Service Portal User Data Manager	ent.repository.ldap.subscriber.manager.security.protocol
Enterprise Service Portal Service Data Manager	ent.repository.ldap.service.manager.security.protocol

If there is no comment character at the beginning of the line, the property is already enabled.

3. Set the server port property (as listed in Table 13) to the value supported for the LDAPS connection.

Table 13: Server Port Properties for SAE Components

SAE Component	Server Port Property
Configuration Manager	Conf.directory.port
User Data Manager	UserDataSource.repository.ldap.server.port
Equipment Data Manager	UserCacheDataSource.repository.ldap.server.port
Service Data Manager	ServiceDataSource.repository.ldap.server.port
License Manager	LicenseMgr.repository.ldap.server.port
Enterprise Service Portal User Data Manager	ent.repository.ldap.subscriber.server.port
Enterprise Service Portal Service Data Manager	ent.repository.ldap.service.server.port

   For LDAPS connections, the default port number is 636.

4. Save the configuration.

Disabling LDAPS Communication for SAE Components

To disable an LDAPS connection for that component:

1. Open the configuration for the security properties for the component. See Table 11.

2. Add a comment character before the component's security protocol property. See Table 12.

3. Set the server port property (as listed in Table 13) to the value supported for the LDAP connection.

   For LDAP connections, the default port number is 389.

4. Save the configuration.

Chapter 9

Integrating Data with the LDAP Directory

This chapter describes how developers can create data integrators that read data from a service provider's storage medium, and write the data in a format that complies with the SDX LDAP schema to a directory. The chapter contains the following sections:

- Overview of Data Integration on page 93
- Getting Help with Data Integration on page 95
- Installing the Data Integration Suite on page 96
- Planning Data Integration on page 96
- Developing Data Integrators on page 96
- Configuring Data Integrators on page 97
- Executing Data Integration on page 102
- Examples of Data Integrators on page 102

Overview of Data Integration

The SRC software uses data that complies with the SDX LDAP schema and that is stored in one of the supported directories. Service providers, however, may store data that they want to use for the SRC software in another storage medium, such as a database. You can develop data integrators that read your data from a storage medium, and write the data to a directory for use with the SRC software. A typical use of a data integrator is to read information about virtual private networks (VPNs) from your company's proprietary database and to write the data to a directory for use with the SRC software.

The data integration suite comprises a set of processors that perform different data management tasks. You can use combinations of these processors to achieve the data integration that you require. The processors use Extensible Markup Language (XML) documents to perform the data transfer. There are three types of processors:

- Readers, which read data from a storage medium and write the data to an XML document.
- Extensible Stylesheet Language Transformation (XSLT) transformers, which convert an XML document into a different XML document that another processor can use.
- Writers, which read data from an XML document in memory and write data to a storage medium.

Table 14 lists the processors, their functions, and associated document type definitions (DTDs). You can see the associated DTDs in the SRC software distribution in the folder *SDK/dtd/dataint* or on the Juniper Networks Web site at

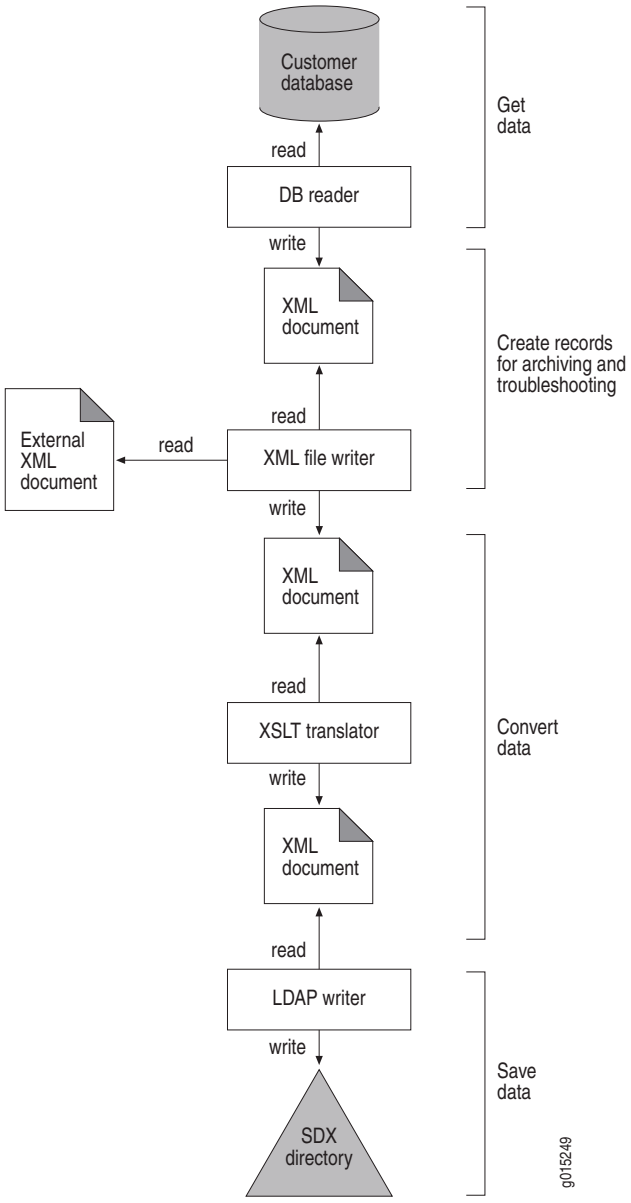
http://www.juniper.net/techpubs/software/management/sdx

Table 14: Processors for Data Managers

Processor	Function	Associated DTDs
Database Reader	Reads data from a database and writes the data to an XML document.	This processor does not have an associated DTD. Instead, it uses a DTD scheme with tags that correspond to the attribute names in the SQL queries that you write.
LDAP Reader	Reads data from an LDAP-compliant directory and writes the data to an XML document.	This processor uses two DTDs: ■ The input DTD, *LDAPReader_input.dtd,* describes a query to a directory. ■ The output DTD, *LDAPReader_output.dtd* describes the result of a query to the directory.
XML File Reader	Reads data from an XML document and passes the data to the next processor.	None
Enterprise Audit File Reader	Reads data from the Enterprise Service Portal audit plug-in log and writes the data to an XML document with a DTD specific to the enterprise audit plug-in log.	This processor uses the DTD *EntAuditFileReader_output.dtd*, which describes the result of a query to an enterprise audit log.
XML File Writer	Reads data from an XML document, writes the data to an external XML document, and returns the original XML document. Use this processor to create external XML documents of data either for record-keeping purposes or for troubleshooting the data integration.	None
XSLT Translator	Takes the XML document it receives from the preceding processor and writes it to an XML document that another processor can read. For this processor, you must customize an XSLT file that maps the input XML document to the output XML document.	None
LDAP Writer	Reads data from an XML document and writes the data to an LDAP directory.	This processor uses the DTD *LDAPWriter_input.dtd*, which describes a set of directory updates.

Figure 12 illustrates how you can use a combination of the processors to transfer data from your database to a directory that contains SDX data. For more detailed examples, see *Examples of Data Integrators* on page 102.

Figure 12: Transferring Data from a Database to the SDX Directory



Getting Help with Data Integration

Planning data integration and developing data integrators are tasks that should be undertaken by developers with knowledge of XML, XSLT, databases, and directories. For assistance with these tasks, consult Juniper Networks Professional Services.

Installing the Data Integration Suite

For information about installing the data integration suite, see *SDX Getting Started Guide, Chapter 28, Installing the SRC Software on a Solaris Platform*. After you have installed the data integration package, copy the Java Database Connectivity (JDBC) drivers for any databases from which you will extract data to the folder *opt/UMC/datint/lib/*.

Planning Data Integration

Before you develop data integrators, create a plan for the data integration. To do so:

1. Determine the data that you want to transfer, the sources of the data, and the type of transfer that you want to perform.

   For example, you might want to transfer data about subscribers, networks, and VPNs from your database to the SDX directory. You might also want to create XML documents of the original data that you can use for troubleshooting as you create the data manager.

2. Determine how to divide the transfer process into smaller data transfers that limit the size of the query result and minimize the time that the LDAP Writer spends interacting with the directory.

   For example, you might create three data integrators, each of which manages data for one of the following areas:

   - Subscribers
   - Network
   - VPNs

3. Identify the processors that you need to perform the data transfer, and the properties that you will use for each processor.

   For a detailed description of each processor and its properties, see the online documentation in the SRC software distribution in *SDK/doc/dataint*.

Developing Data Integrators

To develop a data integrator:

1. For each XSLT transformer that you use, create an XSLT file that maps the structure of the input XML document to an output XML document that the next processor in the chain can read.

   You can view examples of XSLT files in the folder */opt/UMC/datint/xslt*. For more information about these examples, see *Examples of Data Integrators* on page 102.

2. If you are using the Database Reader processor, write an SQL query that the processor uses to obtain information from the database.

3. Create a property file that defines the properties for the data integrator (see *Configuring Data Integrators* on page 97).

4. Create a script that calls the data integrator.

5. (Optional) Configure a utility, such as a crontab file, to run this script at a defined time.

   The script or crontab file can run the data integrator either once with a single property file or multiple times with different property files. For an example of a script that calls a data integrator, see the file *vpndatamgt* in */opt/UMC/datint/etc*.

Configuring Data Integrators

To configure a data integrator, you must create a property file that contains specific properties for the overall data transfer and for the individual data processors. When you create a property file, you must:

1. Define logging properties by using the standard property names and values for SRC logging. To define the logging properties, use the following format:

   Logger.<groupName>.<propertyName>=<value>

   - <groupName> —Name of group for this log
   - <propertyName> —Name of property
   - <value> —Value of property

   For detailed information about configuring logging properties, see *SDX Monitoring and Troubleshooting Guide, Chapter 4, Configuring Logging for SRC Components on a Solaris Platform*.

2. Define properties for the individual processors. The properties that you must or can configure depend on the particular processor. To define properties for the processors, use the following format:

   Processor.<processorName>.<propertyName>=<value>

   - <processorName> —Name of processor that you define; each processor in a property file must have a unique name
   - <propertyName> —Name of property; may comprise several text strings separated by dots
   - <value> —Value of property

   To define properties for each processor, see:

   - Defining Properties for the Database Reader on page 98
   - Defining Properties for the LDAP Reader on page 99
   - Defining Properties for the XML File Reader on page 100
   - Defining Properties for the Enterprise Audit File Reader on page 100
   - Defining Properties for the XML File Writer on page 101

- Defining Properties for the XSLT Translator on page 101
- Defining Properties for the LDAP Writer on page 101

For detailed information about each processor, see the online documentation in the SRC software distribution in *SDK/doc/dataint*.

3. Define the order in which the processors are called. To define the order in which the processors will be executed, enter one statement in the following format:

   Processor.chain=<comma-separated list of names of processors in order>

   For example:

   Processor.chain=dbreader,toldap,xmlfilewriter,ldapwriter

Defining Properties for the Database Reader

You must define the following properties for this processor:

- Class of the processor
- Data that you want to read from the database

  Processor.dbreader.dbQuery=SELECT <sqlQuery>

  - < sqlQuery > —SQL query that summarizes and processes data from the database

- Class of the JDBC driver

  Processor.dbreader.driverClass=<driverClass>

  - < driverClass > —Class of JDBC driver

- URL of the database

  Processor.dbreader.dbURL=<databaseURL>

  - < databaseURL > —URL of the database

- Username and password for logging into the database

  Processor.dbreader.user=<username>
  Processor.dbreader.password=<password>

  - < username > —Username that the database uses to authenticate the processor
  - < password > —Password that the database uses to authenticate the processor

- Output format, Document Object Model (DOM) for the query result

  Processor.dbreader.out=dom

- Inclusion of data in the XML document that the processor returns

  Processor.dbreader.genData=true

- Any optional properties that you require

  For information about optional properties for this processor, see the online documentation in the SRC software distribution in *SDK/doc/dataint*.

The following example is a property file for this processor:

```
# Database Reader
Processor.dbreader.class=net.juniper.smgt.ent.datamgt.reader.DBReader
Processor.dbreader.driverClass=org.gjt.mm.mysql.Driver
Processor.dbreader.dbURL=jdbc:mysql://127.0.0.1:3306/vpn
Processor.dbreader.user=admin
Processor.dbreader.password=secret
Processor.dbreader.genData=true
Processor.dbreader.out=dom

# The SQL query
Processor.dbreader.dbQuery=SELECT vpn_ownership.vpn_id,
vpn_ownership.vpn_owner,vpn_sites.router_name,vpn_sites.interface_nameFROM
vpn_ownership, vpn_sites where vpn_ownership.vpn_id=vpn_sites.vpn_id

# XML element names
Processor.dbreader.elname.database=database
#Processor.dbreader.elname.record=record
```

Defining Properties for the LDAP Reader

This processor obtains the query it performs as an XML document from the previous processor in the chain, and you do not need to define the query. You must, however, define the following properties for this processor:

- Name of the class of the processor

- DES properties in the format

  Processor.<processorName>.<desProperty>=<value>

  - <processorName>—Name of processor that you define; each processor in the same property file must have a unique name

  - <desProperty>—Name of the DES property

  - <value>—Value of the DES property

  For information about DES properties and values, see *SDX Getting Started Guide, Chapter 32, Distributing Directory Changes to SRC Components on a Solaris Platform*.

- Any optional properties that you require

  For information about optional properties for this processor, see the online documentation in the SRC software distribution in *SDK/doc/dataint*.

The following example is a property file for this processor:

```
Processor.ldapreader.class=net.juniper.smgt.ent.datamgt.reader.LDAPReader
Processor.ldapreader.java.naming.provider.url = ldap://127.0.0.1/
Processor.ldapreader.java.naming.security.principal = cn=umcadmin,o=umc
Processor.ldapreader.java.naming.security.credentials = admin123
Processor.ldapreader.continuous=true
Processor.ldapreader.java.naming.provider.url = ldap://127.0.0.1/
```

Defining Properties for the XML File Reader

You must define the following properties for this processor:

- Name of the class of the processor
- In the following format, the XML document from which this processor reads data:

  Processor.<processorName>.XMLFileName= <path>

  - <processorName>—Name of the processor that you define; each processor in the same property file must have a unique name
  - <path>—Path to XML document relative to the folder */opt/UMC/datint*

The following example is a property file for this processor:

```
Processor.xmlfilereader.class=net.juniper.smgt.ent.datamgt.reader.XMLFileReader
Processor.xmlfilereader.XMLFileName=var/log/dbout.xml
```

Defining Properties for the Enterprise Audit File Reader

You must define the following properties for this processor:

- Name of the class of the processor
- In the following format, the log file from which this processor reads data:

  Processor.<processorName>.auditFileName=<path>

  - <processorName>—Name of the processor that you define; each processor in the same property file must have a unique name
  - <path>—Path to the XML document relative to the folder */opt/UMC/datint*
- Any optional properties that you require

  For information about optional properties for this processor, see the online documentation in the SRC software distribution in *SDK/doc/dataint*.

The following example is a property file for this processor:

```
Processor.auditfilereader.class=net.juniper.smgt.ent.datamgt.reader.EntAuditFileReader
Processor.auditfilereader.auditFileName=ent_audit.log
Processor.auditfilereader.filter=(Action=Unexport-VPN)
```

Defining Properties for the XML File Writer

You must define the following properties for this processor:

- Name of the class of the processor
- In the following format, the XML document to which the processor writes data:

  Processor.xmlfilewriter.XMLFileName=<path>

  - < path > —Path to the XML document relative to the folder */opt/UMC/datint*

The following example is a property file for this processor:

```
Processor.xmlfilewriter.class=net.juniper.smgt.ent.datamgt.filter.XMLFileWriter
Processor.xmlfilewriter.XMLFileName=var/log/ldapout.xml
```

Defining Properties for the XSLT Translator

You must define the following properties for this processor:

- Name of the class of the processor
- In the following format, the XSLT file that the processor uses

  Processor.<processorName>.XSLTFileName=<path>

  - < processorName > —Name of the XSLT translator
  - < path > —Path to XSLT file relative to the folder */opt/UMC/datint*

The following example is a property file for this processor:

```
Processor.toabstract.class=net.juniper.smgt.ent.datamgt.filter.XSLTTranslator
Processor.toabstract.XSLTFileName=xslt/vpn.xslt
```

Defining Properties for the LDAP Writer

You must define the following properties for this processor:

- Name of the class of the processor
- DES properties in the format

  Processor.ldapwriter.<desProperty>=<value>

  - < desProperty > —Name of the DES property
  - < value > —Value of the DES property

  For information about DES properties and values, see *SDX Getting Started Guide, Chapter 32, Distributing Directory Changes to SRC Components on a Solaris Platform*.

- Any optional properties that you require

  For information about optional properties for this processor, see the online documentation in the SRC software distribution in *SDK/doc/dataint*.

The following example is a property file for this processor:

```
Processor.ldapwriter.class=net.juniper.smgt.ent.datamgt.filter.LDAPWriter
Processor.ldapwriter.java.naming.provider.url = ldap://127.0.0.1/
Processor.ldapwriter.java.naming.security.principal = cn=umcadmin,o=umc
Processor.ldapwriter.java.naming.security.credentials = admin123
Processor.ldapwriter.updateRateLimit=3
Processor.ldapwriter.continuous=true
```

Executing Data Integration

To execute data integration, run the script that calls the property files, or configure a utility, such as a crontab file, to run this script at a defined time.

Examples of Data Integrators

After you install the data integration suite, you can access two data integrators that we have developed:

- VPN Directory Updater
- VPN Subscription Deactivator

The following sections describe each of these data integrators. You can also examine the following files to see how these data integrators were developed.

- Property files in */opt/UMC/datint/etc*
- XSLT files in */opt/UMC/datint/xslt*
- The script **vpndatamgt** in */opt/UMC/datint/etc*, which calls both these data integrators

Example: VPN Directory Updater

VPN Directory Updater is a sample data integrator that reads data about VPNs from a database and writes to a directory the data in a format that meets the SDX LDAP schema. If you want to use VPN Directory Updater, you must customize it for your specific application. At the very least, you need to customize the SQL queries for your database.

VPN Directory Updater works as follows:

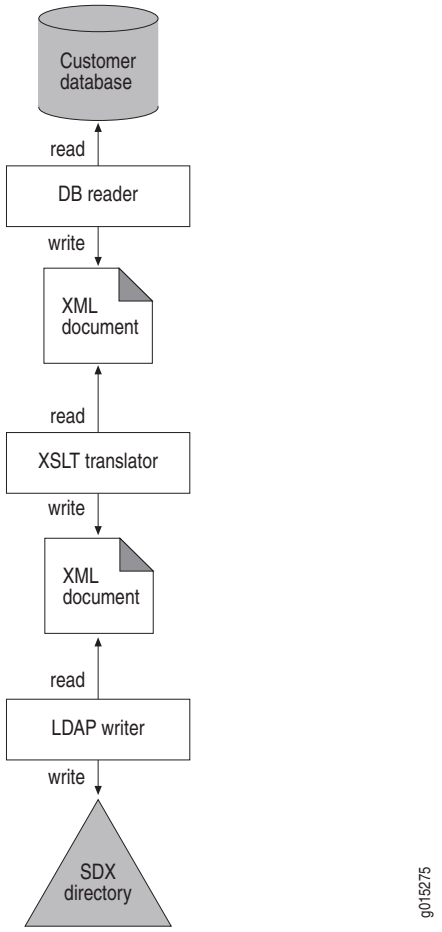
1. Database Reader submits SQL queries to a database, obtains the result of the query, and converts the result to an XML document.

2. XSLT Translator takes the XML document produced by Database Reader and converts it to an XML document that describes a set of directory updates.

3. LDAP Writer uses the XML document generated by XSLT Translator to update the directory.

Figure 13 illustrates this process.

Figure 13: VPN Directory Updater Operation



Example: VPN Subscription Deactivator

If an IT manager cancels the export of a VPN at the same time that an extranet client activates a subscription to this VPN, there is a remote possibility that the Enterprise Manager portal will maintain the active, but invalid, subscription. VPN Subscription Deactivator deactivates this type of invalid VPN subscription.

You can use VPN Subscription Deactivator without modifications. For information about using this data integrator, see *SDX Subscribers and Subscriptions Guide, Chapter 24, Adding VPNs from JUNOS Routing Platforms*.

VPN Subscription Deactivator works as follows:

1. Enterprise Audit File Reader finds events of type unexport-vpn in the log for the Enterprise Audit Plug-In and converts the events to an XML document.

2. The XSLT Translator *tovpnsubquery* performs the following actions:

    a. Reads the XML document from the Enterprise Audit File Reader and uses it to identify extranet clients for whom export of a VPN was canceled.

    b. Uses the XSLT file *tovpnsubquery.xslt* to generate LDAP queries to obtain subscriptions, imported extranets, and VPNs owned for these extranet clients.

    c. Writes the queries to an XML document.

3. LDAP Reader performs the following actions:

    a. Reads the XML document from the XSLT Translator *tovpnsubquery*.

    b. Submits the queries to the directory.

    c. Obtains the results of the queries from the directory.

    d. Writes the results to an XML document.

4. The XSLT Translator *chkghostvpnsub* performs the following actions:

    a. Reads the XML document from the LDAP Reader.

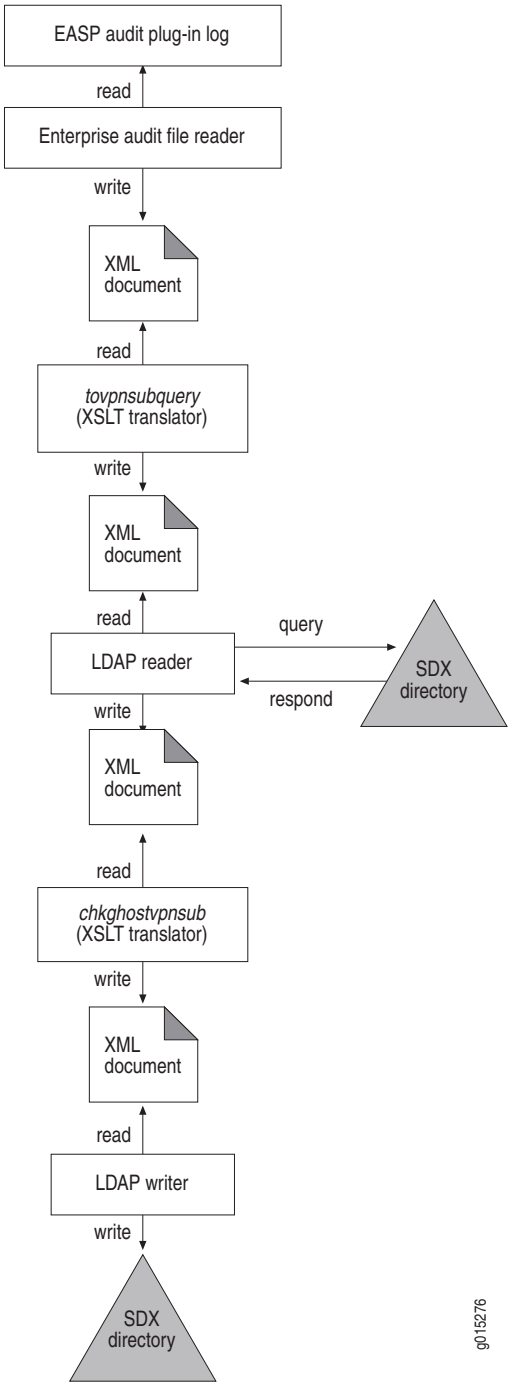
    b. Uses the XSLT file *chkghostvpnsub.xslt* to find in the XML document VPN subscriptions that are still active even though the export of the VPN has been canceled.

    c. Generates an XML document that contains LDAP updates to deactivate the invalid subscriptions.

5. LDAP Writer uses the XML documents generated by Enterprise Audit File Reader and LDAP Reader to update the directory.

Figure 14 illustrates this process.

Figure 14: VPN Subscription Activator



Chapter 10

Access Control Scheme

Each of the SRC components has an entry in the directory under *ou = components*, *o = operators*, *o = umc*. Service providers can establish a multilayered access control scheme for operators. For instance, a network operator might be able to write new objects only under the folder *o = network*. The operator entries are subordinates of *o = operators, o = umc*. This chapter contains the following sections:

- Directory Configuration on page 107
- Directories on page 108
- User Class on page 108
- Permissions on page 108
- Access Controls on page 109
- Directory-Specific Access Control Implementation on page 120

Directory Configuration

During configuration of the directory, the following entries for components and operators are created:

- bind DN for SSP: *cn = ssp, ou = components, o = operators, o = umc*
- bind DN for RADIUS: *cn = radius, ou = components, o = operators, o = umc*
- bind DN for POM: *cn = pom, ou = components, o = operators, o = umc*
- bind DN for directory eventing: *cn = des, ou = components, o = operators, o = umc*
- bind DN for workflow: *cn = workflow, ou = components, o = operators, o = umc*
- bind DN for object state machine: *cn = osm, ou = components, o = operators, o = umc*
- bind DN for system management: *cn = sysman, ou = components, o = operators, o = umc*
- bind DN for SDX operators: *cn = ssc-operator, o = operators, o = umc*

- bind DN for network operators: *cn = network-operators, o = operators, o = umc*
- bind DN for service operators: *cn = service-operator, o = operators, o = umc*
- bind DN for subscriber operators: *cn = subscriber-operator, o = operators, o = umc*

Directories

Directories specify the access rights for certain users to particular information in the directory, whereas other users might not receive any rights to that information. The access rights are defined through access control lists. Using access control lists, you can define permissions to the following targets:

- Entire directory content
- Particular subtree in the directory
- Objects that match a given search filter
- Specific object in the directory

The objects that are part of the target can be protected on an entry level and on an attribute level.

User Class

The access control lists specify the user class from which the items are protected. The user class can be one of the following:

- Specific user
- Members of a specific group
- All entries of a subtree
- Users that match a given search filter
- All users
- This entry (for self-administration)

Permissions

You must set permissions for the target. The following permissions are available:

- Add
- Search
- Compare
- Filter match

- Modify (write)
- Read
- Remove (delete)
- Rename

You can grant or deny these permissions. Deny takes precedence over grant.

Access Controls

Access Controls for the Entire Tree

A client who accesses the directory without binding to it does not have any access rights. All clients who bind with the credentials of an SRC component or an operator are members of the SSC-component-operator group and by default have the following access rights:

- No access to the subtree o = *Operators, o = umc*
- Read access to the remaining directory tree, including the operational attributes creationTimeStamp and modifyTimeStamp
- No read and compare rights for any userPassword values

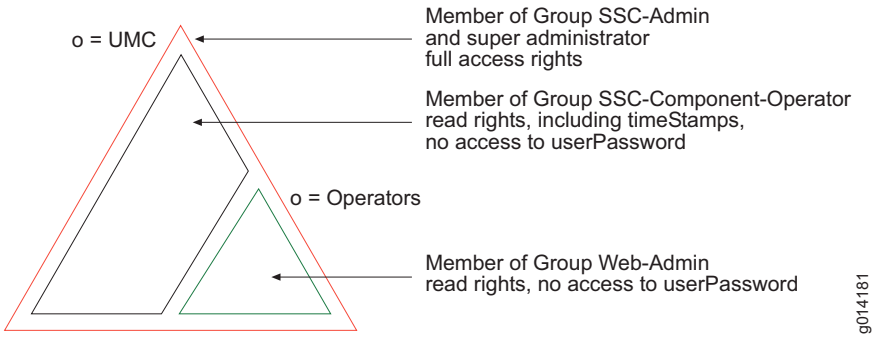
Clients binding with the Apache DN or a member of the WebAdmin group do have read and search permissions in the subtree *o = Operators, o = umc*:

- Read access for all user attributes
- No read and no filter match permissions for the attribute userPassword

Members of the WebAdmin group are allowed to administer the SAE through the SAE Web Administration pages.

The members of the SSC_Admin group and the super-administrator have access rights to the entire tree.

Figure 15: Access Rights for the UMC Tree

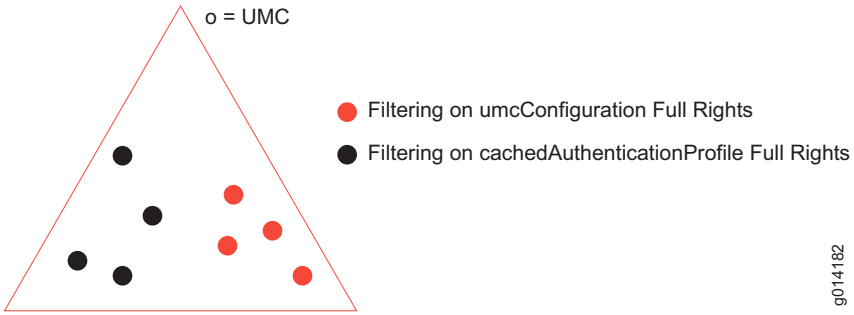


Access Controls Against Objects from Type cachedAuthentication Profile and UmcConfiguration

The SAE binds as *cn = ssp, ou = components, o = operators, o = umc* against the directory and needs to have full access rights for the entries from the type object class cachedAuthenticationProfile and umcConfiguration.

It is easier to implement the cached entries through the targets of the two subtrees *(o = AuthCache, o = umc and o = UserProfilesCache, o = umc)*.

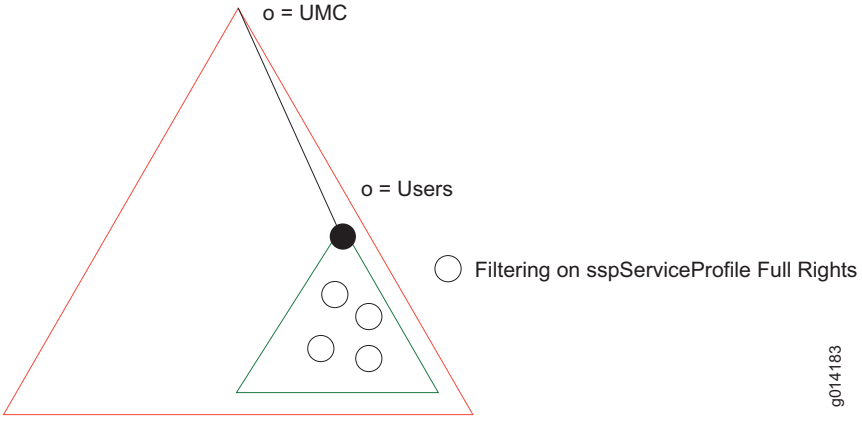
Figure 16: Access Rights Against cachedAuthenticationProfile and umcConfiguration Objects



Access Controls Against sspServiceProfile

In addition to the previously discussed access rights, the SAE requires full access against objects from the tree sspServiceProfile.

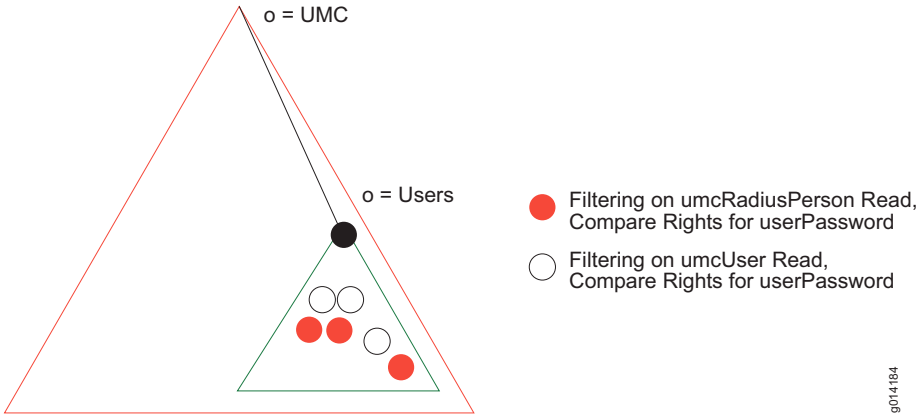
Figure 17: Access Controls Against sspServiceProfiles in the User Subtree



Access Controls Against umcRadius Person and umcUser

The SAE requires read access to the userPassword attribute for entries from type umcRadiusPerson and umcUser.

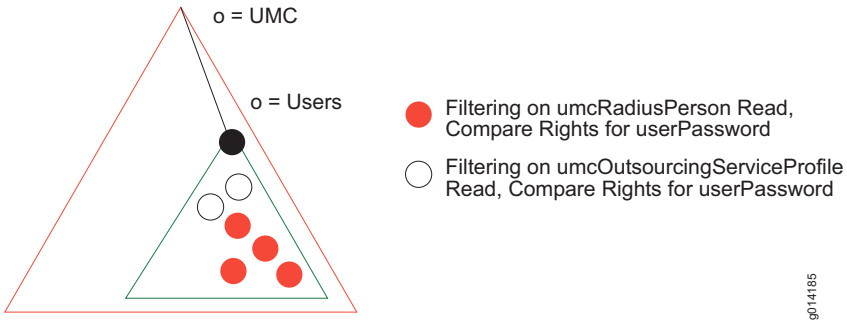
Figure 18: Access Rights Against umcRadiusPerson and umcUser



Access Controls Against RADIUS Profiles

RADIUS requires read access to the userPassword attribute in entries from umcRadiusPerson to authenticate requests of a subscriber, and from umcOutsourcingServiceProfile to determine the tunnel parameter for a Layer 2 Tunneling Protocol (L2TP) outsourcing scenario. The RADIUS server binds with the credentials of *cn = radius, ou = components, o = operators, o = umc*.

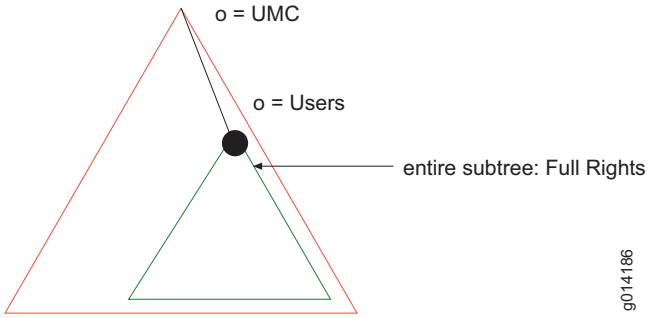
Figure 19: Access Rights Against umcRadiusPerson and umcOutsourcingServiceProfile Objects



Access Controls Against the Policy Subtree

The policy management component uses the credentials of *cn = pom, ou = components, o = operators, o = umc* and requires the following set of access rights for the policy subtree. It needs to perform add, delete, and modify operations on all policy and policyFolder objects in the *o = Policies, o = umc* subtree.

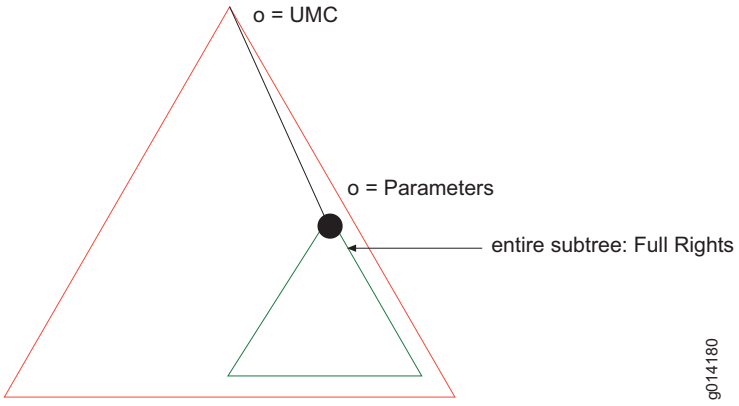
Figure 20: Policy Rights Against All Objects in the *o=Policies,o=umc* Tree



Access Controls Against the Parameter Subtree

The policy management component requires the following set of access controls for the parameter subtree. It needs to perform add, delete and modify operations on all objects in the *o = Parameter, o = umc* subtree.

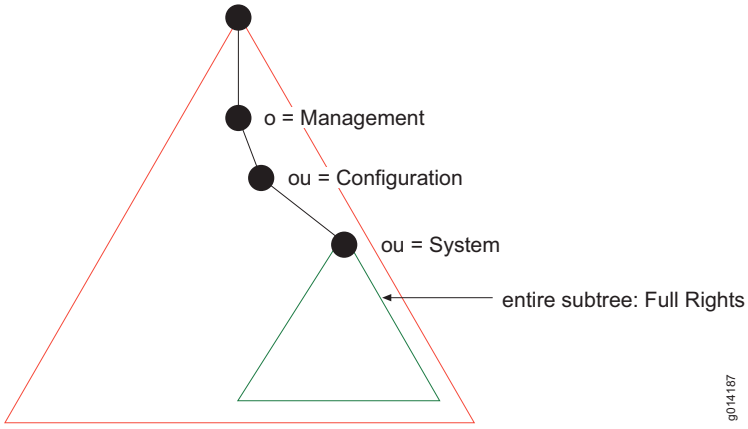
Figure 21: Access Rights Against All Objects in the Tree *o=Parameters, o=umc*



Access Controls for System Management

The system management component binds as *cn = sysman, ou = components, o = operators, o = umc* and requires full access rights for the subtree *ou = SystemManagement, o = Configuration, o = Management, o = umc*.

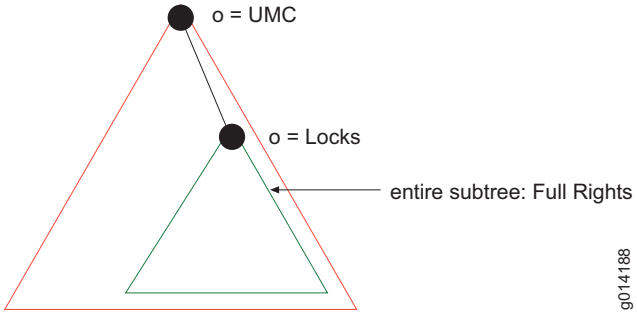
Figure 22: Access Rights for System Management



Access Controls Against the Lock Subtree

The object state manager component requires full access rights to the subtree *o = Locks, o = umc*. This component uses the credentials of *cn = osm, ou = components, o = operators, o = umc* to bind against the directory.

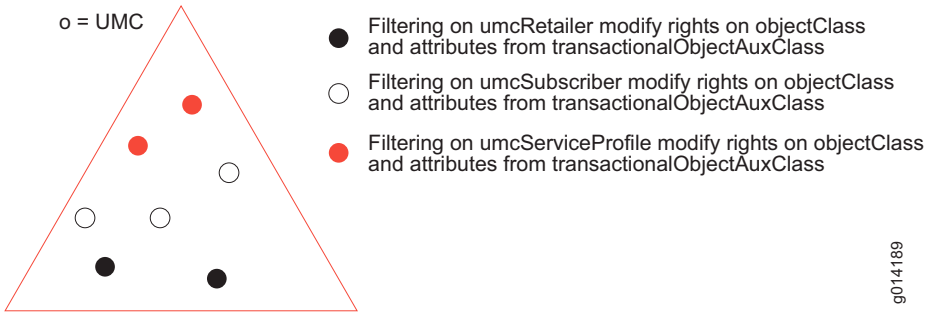
Figure 23: Access Rights Against the Entire *o=Locks,o=umc* Subtree



Access Controls Against Subscriber, Retailer, and Service Profiles

The workflow component needs to flag objects that are in a transactional state. Those objects can be any umcSubscriber, umcRetailer, or umcServiceProfile object. The component must have modify rights on those target objects and write access to all attributes that are part of the auxiliary class transactionalObjectAuxClass, as well as the attribute objectClass. The workflow component binds with the credentials of *cn = workflow, ou = components, o = operators, o = umc* against the directory.

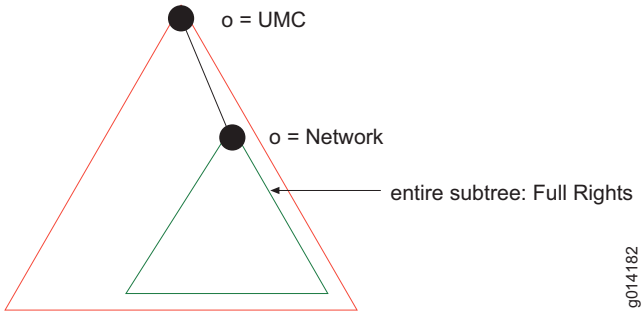
Figure 24: Access Rights Against umcSubscriber, umcRetailer and umcServiceProfile Objects



Access Controls Against the Network Subtree

The network operator is allowed to administer only objects within the subtree *o = Network, o = umc* and bind against the directory using the credentials of *cn = network-operator, o = operators, o = umc*.

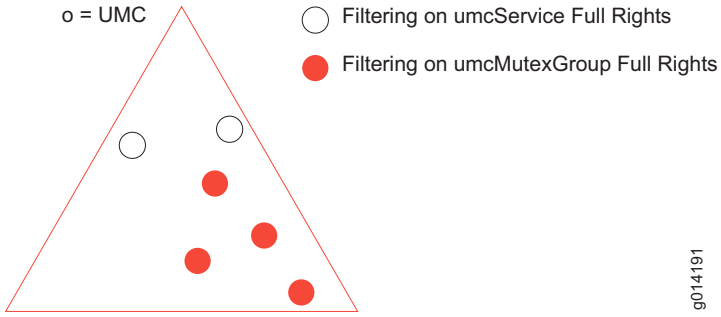
Figure 25: Access Rights Against the Entire *o=Network,o=umc* Subtree



Access Controls Against Services and Mutex Group Objects

The service operator requires full access rights for umcService objects, as well as for umcMutexGroup objects. These objects are subordinates of the entries *o = Services, o = umc* and *o = Scopes, o = umc*. The service-operator binds with the DN *cn = service-operator, o = operators, o = umc* against the directory.

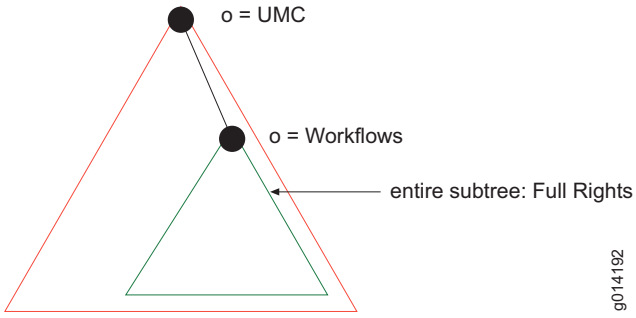
Figure 26: Access Rights Against umcService and umcMutexGroup Objects



Access Controls Against the Workflow Subtree

Workflow operators manage all workflow objects within the subtree *o = Workflows, o = umc*. Therefore, these operators require full access rights for the subtree *o = Workflows, o = umc*. Such operators use the credentials of *cn = workflow-operator, o = operators, o = umc* against the directory.

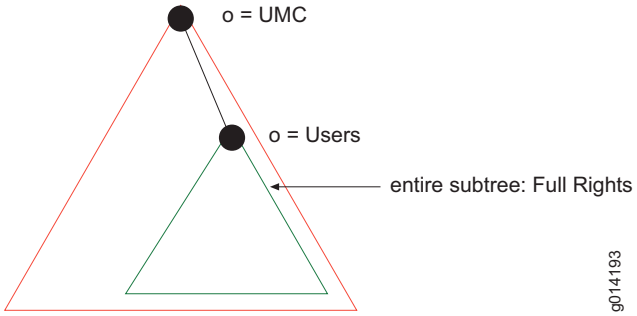
Figure 27: Access Rights Against the Entire *o=Workflows, o=umc* Subtree



Access Controls Against the User Subtree

Subscriber operators are responsible for the entire *o = users, o = umc* subtree and require full access rights. The subscriber operator uses the credentials of the entry *cn = subscriber-operator, o = operators, o = umc*.

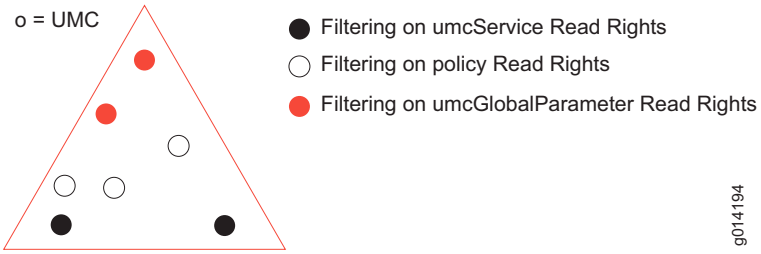
Figure 28: Access Rights Against the Entire *o=users, o=umc* Subtree



Access Controls Against Service, Policy, and Global Parameter Objects

All enterprise managers require read and search rights against objects from the type umcService, policy, and umcGlobalParameter. Those managers bind with their credentials against the directory.

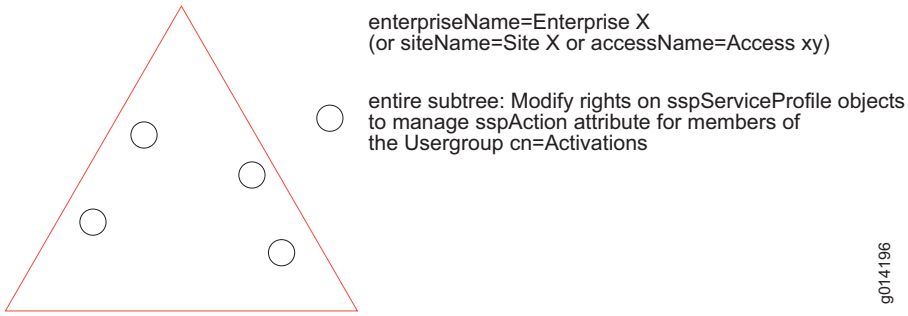
Figure 29: Access Rights Against umcService, Policy, and umcGlobalParameter Objects



Activation Access Rights

Operators who are members of the user group cn = Activations need to be able to change the attribute sspAction to activate or deactivate SSP services in an enterprise, site, or access scope. Figure 30 shows these modify rights.

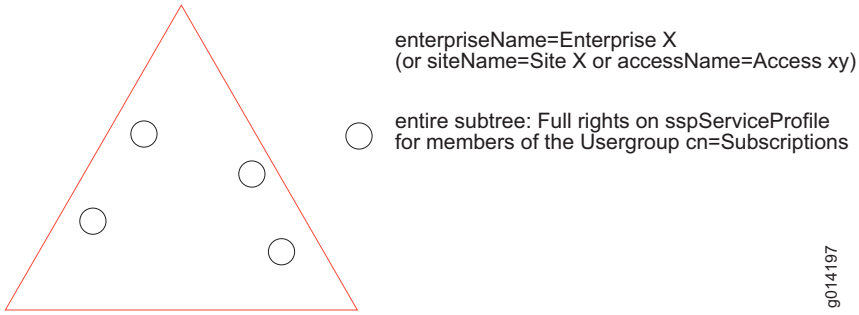
Figure 30: Modify Rights for Activation Managers



Subscription Access Rights

Subscription operators are members of the user group cn = Subscriptions and are able to subscribe and unsubscribe to and from SSP services in their specific scope (that is, enterprise, site, or access). This is the creation and deletion of objects from the type sspServiceProfile. As a result, subscription operators require full access rights to the objects shown in Figure 31.

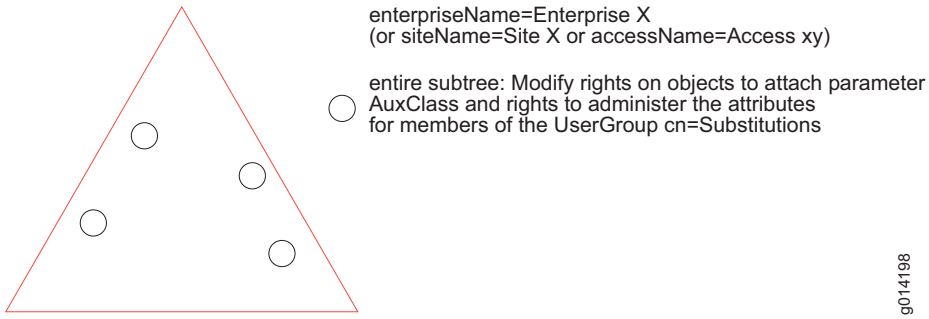
Figure 31: Access Rights for Subscription Managers



Substitution Access Rights

Members of the substitutions user group get the required access rights that grant to attached auxiliary object classes, to objects and modify the attribute type belonging to the auxiliaryclass parameterAuxClass.

Figure 32: Access Rights for Substitution Managers

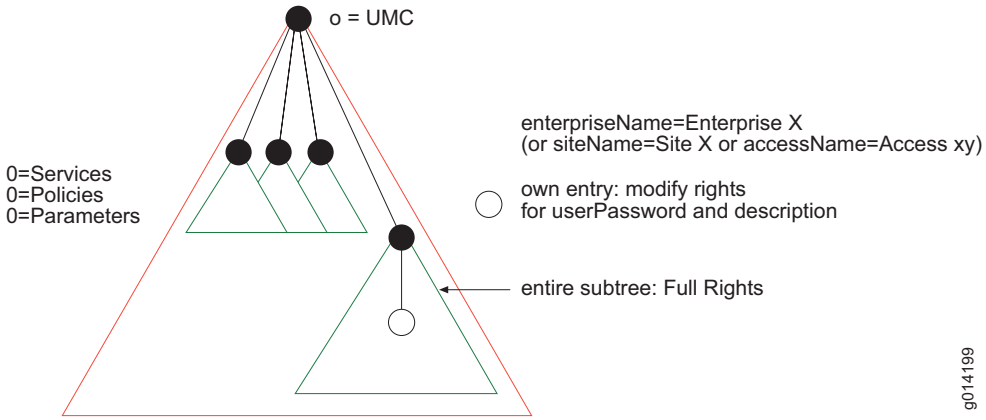


Common Access Rights for All Managers

All enterprise managers (that is, members of the previously mentioned user groups) have the following common rights:

- Read access to the service subtree (*o = services, o = umc*)
- Read access to the policy subtree *(o = policies, o = umc)*
- Read access to the global parameter subtree (*o = parameters, o = umc*)
- Read access to the scope of the manager; that is, enterprise, site, or access read access
- Modify rights to change the user password and description value of its entry

Figure 33: Access Rights for All Managers



Directory-Specific Access Control Implementation

The supported directories (for example, DirX, and Sun ONE) have complex mechanisms for controlling access, depending on the user bound to the directory.

DirX stores the access control lists in subentries that conform to the X.500 standard. You create the access control subentries by using the DirX client dirxcp. These access control subentries are replicated in a shadowing scenario.

Sun ONE stores the access control lists in the directory. Sun ONE extends the standard object class top by the optional attribute aci, which is used to store the access control lists. This means that the access control information (ACI) can be added through LDAP. The aci values are replicated to the slave directory.

DirX Directory Server

DirX access control information is stored in subentries that are from the type *subentry* and *acceSDXontrolSubentry*. These subentries include the information about the target (that is, what is controlled), precedence (that is, higher precedence overwrites lower precedence), and the access control information (that is, prescriptive ACI) that includes the user class (that is, who is affected by the control parameters) and the permissions on entry and attribute level.

Access control subentries can contain many prescriptive ACIs with a list of one or more items to be protected, such as entries and sets of operation or user attributes.

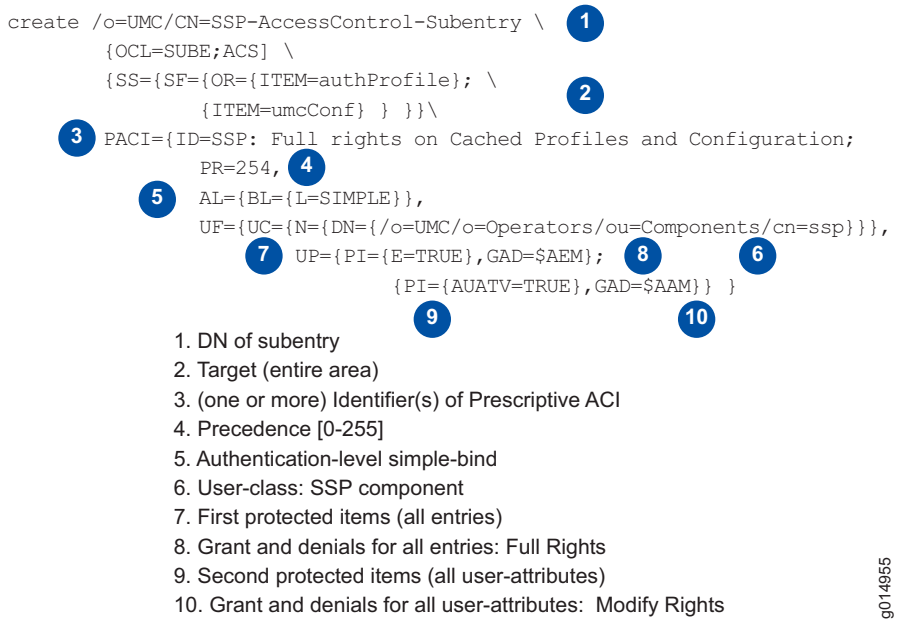
The UMCdirxa package includes a TCL file, called *acldefs.tcl*, which defines the following variables for the permissions:

- DAER—Deny read access on entry level
- AER—Grant read access on entry level
- AEM—Grant full access on entry level
- AEME—Grant modify access on entry level
- DAAR—Deny read access on attribute level
- AAR—Grant read rights on attribute level
- AAM—Grant modify rights on attribute level

The UMCdirxa package includes the file *access.cp,* which sets the access controls for the SRC software.

Figure 34 shows a TCL script with an explanation of the various parts.

Figure 34: Creation of an Access Control Subentry Example in DirX



Sun ONE Directory Server

Access control information is stored in the aci attribute of each directory entry. Because the access control information is stored in the directory, it can be managed by means of LDIF files.

ACIs take the following form:

aci: (<target>) (version 3.0;aci "<name>"; <permission> <bind rule>;)

where

<target> defines the object, attribute, or filter that you are using to define what resource to control access to. The target can be a distinguished name, one or more attributes, and/or a single LDAP filter.

version 3.0 is a required string that identifies the ACI version.

aci "<name>" is a name for the ACI. <name> can be any string that identifies the ACI. The ACI name is required.

<permission> defines the actual access rights and whether they are to be allowed or denied.

<bind rules> identify the circumstances under which the directory login must occur for the ACI to take effect.

The UMCiDSa package includes the LDIF file *access.ldif*, which implements the SDX access control scheme.

Figure 35 shows the LDIF file for implementing the same kind of access level as previously depicted with a Sun ONE directory.

Figure 35: Creation of Access Control List Example in Sun ONE

```
dn: o=UMC
changetype: modify
add: aci
aci:(target="ldap:///o=UMC")(targetattr="*") 1
  (targetfilter="(|(objectClass=cachedAuthenticationProfile) 2
  (objectClass=umcConfiguration))")
  (version 3.0; acl "SSP: enable admin of cahced profiles and configuration"; 3
4 allow (all) userdn = "ldap:///cn=ssp,ou=Components,o=Operators,o=UMC"
5 and (authmethod = "Simple");)
```

1. All user-attributes implicitly included
2. Target (entire area)
3. (one or more) Identifier(s) of Prescriptive ACI
4.Grant write access, which includes the rights auth, compare, read, and search
5. User-class: SSP component

g014957

Part 3

Integrating RADIUS Servers

Chapter 11

Integrating Steel-Belted Radius/SPE Server

The Juniper Networks Steel-Belted Radius/Service Provider Edition (SPE) server is a carrier-grade RADIUS/AAA solution. It provides the reliability, performance, and specialized technology demanded by carriers, wholesalers, and service providers. Use the information in this chapter to integrate the Steel-Belted Radius/SPE server with JUNOSe routers. Refer to the *SRC-PE Release Notes* for information about compatibility of this SRC release with Steel-Belted Radius/SPE server releases. The SRC software does not support the use of RADIUS with JUNOS routing platforms.

This chapter contains the following sections:

- System Requirements for the Steel-Belted Radius Server on page 126
- Installing the Steel-Belted Radius/SPE Software on page 126
- Enabling LDAP Authentication on page 128
- Configuring UDP Ports for Steel-Belted Radius Software on page 129
- Starting the Steel-Belted Radius/SPE Server on page 130
- Stopping the Steel-Belted Radius/SPE Server on page 130
- Extending Dictionary Files with JUNOSe Parameters for the Steel-Belted Radius Server on page 131
- Configuring LDAP Authentication on page 131
- Configuring Directed Authentication on page 138
- Customizing the Authentication Log File on page 139
- Configuring the Steel-Belted Radius/SPE Server and RADIUS Clients on page 139

The SRC software can take advantage of a RADIUS server to authenticate against an LDAP server, which is used to store subscriber and service information, among other information.

System Requirements for the Steel-Belted Radius Server

The Solaris host software package includes:

- The RADIUS server process
- Java-based administration GUI
- A number of dictionary and database files to support various authentication methods

The software package requires:

- Operating system—Solaris 8 and higher
- RAM—At least 64 MB of working memory
- Disk—Depends on external database support; at least 105 MB of hard-disk space
- Browser for GUI—Java-capable browser that understands signed Java applets, such as Netscape 4.08 or later for Solaris

Installing the Steel-Belted Radius/SPE Software

You need the Steel-Belted Radius/SPE software CD and a valid license string. Use the procedure that is appropriate for your installation.

NOTE: For the remainder of the document, we assume that Steel-Belted Radius/SPE is installed in the directory */opt/UMC/SPE*.

Installing the Steel-Belted Radius Software for the First Time

To install the software for the first time:

1. Log in as `root`.
2. Copy files from */cdrom/cdrom0/*Unix to the Sun platforms (for example, to */tmp/funk*), and set your working directory to the directory to which the files were copied.
3. Run the **install.sh** script with the **-all** option. Type:

   sh install.sh -all

   The installation script prompts for the server directory.
4. Type the full pathname:

   /opt/UMC/SPE

   The installation script prompts for the license string.

5. Enter a valid license string.

   The installation script prompts for the type of installation from the following list.

   1. Steel-Belted Radius Enterprise Edition
   2. Steel-Belted Radius Service Provider Edition
   3. Steel-Belted Radius Global Enterprise Edition
   4. Steel-Belted Radius HotSpot Edition

6. Enter selection: [2] Steel-Belted Radius Service Provider Edition

   The installation script prompts for the directory in which the administration user interface should be installed.

7. Type the full pathname:

   /opt/UMC/SPE/radadmin

Installing the Steel-Belted Radius Software over Previous Installations

To install the software over a previous installation:

1. Complete the procedure in the previous section. The **install.sh** script may detect the following items on the machine:

   - RADIUS process already running
   - Steel-Belted Radius/SPE configuration files
   - Steel-Belted Radius/SPE database files

   The **install.sh** script prompts for the following message if the script discovers a running server:

   Server is running with pid <x>
   Stop radius server and unconfig/uninstall before
   Installing new version

2. Change to the installation directory of the Steel-Belted Radius/SPE package. You can find the server directory by typing the following command:

   ps -aef | grep radius

3. Stop the server by typing:

   ./S90radius stop

4. Change to your working directory. Unconfigure the previous installation, which removes the startup script and some entries in the */etc/services* and */etc/inetd.conf* files, which are used by the previously installed Steel-Belted Radius/SPE server. Type:

   sh install.sh -unconfig

5. When prompted, enter the path of the existing server directory.

6. Run the **install.sh** script with the **-all** option again. Type:

   sh install.sh -all

   The script checks for existing Steel-Belted Radius/SPE configuration files. The following prompt appears if files are detected:

   ```
   Previous configuration files exist
   Configuration files exist in <server_directory>
   Do you want to discard them? [n]
   ```

7. If you answer `n`, the previous configuration files are copied into the subdirectory *OLDCONFIG*. Otherwise, the previous files are overwritten.

   The script checks for existing Steel-Belted Radius/SPE database files. The following prompt appears if files are detected:

   ```
   Previous database files exist
   Database files exist in <server_directory>
   Do you want to discard them? [n]
   ```

8. If you answer `n`, the previous configuration files are not overwritten, and the new Steel-Belted Radius/SPE version uses the entire administrative database. Otherwise, the database files are overwritten.

   The script prompts you for the license string.

9. Enter a valid license string.

   The installation script prompts you for the directory in which the administration user interface should be installed.

10. Type the full pathname:

    /opt/UMC/SPE/radadmin

Enabling LDAP Authentication

Use this procedure to enable authentication through the LDAP directories. The SRC software requires that the LDAP be enabled as an external database. The LDAP host is used for authentication.

NOTE: The LDAP is not required for integration with just the JUNOSe router.

To enable an LDAP host as an external database used by the Steel-Belted Radius/SPE server.

1. Log in as `root`.

2. Return to the working directory (directory into which the installation files were originally copied; for example, */tmp/funk*).

3. Unconfigure the initial configuration of Steel-Belted Radius/SPE by running **install.sh** script with the **-unconfig** option. Enter the server directory.

```
# sh install.sh -unconfig
Enter server directory [<working-directory>/radius]: /opt/UMC/SPE
Removing  /etc/rc2.d/S90radius /etc/rc2.d/K90radius
Removing RADIUS entries from /etc/services
Removing RADIUS entries from /etc/inetd.conf
kill -HUP 124
Unconfig completed.
```

4. Configure Steel-Belted Radius/SPE with the external database by running **install.sh** with the **-config** option. You must enter the server directory again. In addition, you must select LDAP as the external database, and you must enter the path */opt/UMC/SPE* as the location of the LDAP libraries. In the following example, no SNMP support is configured (see the Steel-Belted Radius/SPE server manuals for more information about SNMP support).

```
# sh install.sh -config
Enter server directory [[<working-directory>/radius]: /opt/UMC/SPE
Creating S90radius.
Setting the default radius directory /opt/UMC/SPE
Do you want to configure SNMP? [n]: n
Do you want to configure for use with External SQL Databases? [n]: n
Do you want to configure LDAP? [n]: y
Enter path for LDAP library files. [/usr/lib/]: /opt/UMC/SPE
Configuration of LDAP complete. Copying S90radius to /opt/UMC/SPE
Creating link.
Radius server configuration completed. Configuring admin...
Modifying  /etc/services  ...
Modifying  /etc/inetd.conf  ...
kill -HUP 133
Admin configuration completed.
```

5. Copy the dictionary and vendor files (*dictiona.dcm*, *juniper.dct* and *vendor.ini*) for the JUNOSe release from the folder *steel_belted_radius* in the SRC software distribution, into the installation directory (*/opt/UMC/SPE*).

Configuring UDP Ports for Steel-Belted Radius Software

The transaction-based RADIUS protocol uses two UDP ports: one for authentication packets and one for accounting packets. You must configure the ports on both sides—the Steel-Belted Radius/SPE server and the RADIUS clients (SRC software and JUNOSe router). For information about RADIUS client/server configuration, see *Configuring the Steel-Belted Radius/SPE Server and RADIUS Clients* on page 139.

The officially assigned UDP port numbers are:

- 1812 for authentication
- 1813 for accounting

Early deployments of RADIUS used 1645/udp for authentication packets and 1646/udp for accounting packets.

The (ports) section of the RADIUS configuration file *radius.ini* allows you to set the UDP ports used for authentication and accounting and the UDPAuthPort and UDPAcctPort fields for port assignment. You can specify more than one port that the SPE server is listening to; for example:

- UDPAuthPort = 1812
- UDPAuthPort = 1645
- UDPAcctPort = 1813
- UDPAcctPort = 1646

If no port settings are present in the *radius.ini* file, the SPE server attempts to read the port numbers associated with the RADIUS servers from the */etc/services* file.

If no port settings are present in the *radius.ini* file and no RADIUS services are defined in the */etc/services* file, the SPE server listens to UDP ports 1645 and 1812 for authentication and UDP ports 1646 and 1813 for accounting.

Starting the Steel-Belted Radius/SPE Server

To start the RADIUS server:

1. Change to the *<server-directory> /opt/UMC/SPE*.
2. Enter:

   ./S90radius start

   During startup, the RADIUS server binds to the LDAP server, which requires that the LDAP server be running before the RADIUS server is started. The RADIUS process is automatically started whenever the Solaris host is booted.

Stopping the Steel-Belted Radius/SPE Server

To stop the RADIUS server:

1. Change to the *<server-directory> /opt/UMC/SPE*.
2. Enter:

   ./S90radius stop

Extending Dictionary Files with JUNOSe Parameters for the Steel-Belted Radius Server

In addition to supporting the standard RADIUS attributes, JUNOSe routers support JUNOSe-specific attributes. You must replace a file to introduce JUNOSe-specific attributes to the Steel-Belted Radius server. Replacing this file is necessary to complete both the Steel-Belted Radius–JUNOSe router integration and the Steel-Belted Radius–JUNOSe router–SRC integration.

To extend dictionary files with JUNOSe parameters, replace the *juniper.dct* dictionary file with the ERX RADIUS Dictionary file. To locate the ERX RADIUS Dictionary file, see the JUNOSe software documentation for the supported release on the Juniper Networks Web site at

http://www.juniper.net/techpubs/software/

The Juniper Networks dictionary file is included as part of the SRC installation media.

Configuring LDAP Authentication

The SRC software assumes that all RADIUS authentications are performed against the SDX LDAP directory. The information in this section also applies to the Steel-Belted Radius/SPE server integration with a JUNOSe router, provided that the Steel-Belted Radius/SPE server uses an LDAP server as an external database for authentication.

Integration of the JUNOSe-specific attributes, such as primary Domain Name System (DNS) and virtual router, must be performed. Steel-Belted Radius/SPE server supports such an external authentication method by using several configuration files.

These files tell the RADIUS server:

- How the RADIUS server communicates with an external database (LDAP)
- How the RADIUS server queries the external database for authentication
- How the RADIUS server formulates the response from the query result

You configure LDAP authentication by modifying properties in the *ldapauth.aut* file, which is located in the server directory (*opt/UMC/SPE*). If you do not specify options in this file, the SPE assumes the default values. You can also view a sample *ldapauth.aut* file in the SRC software distribution in the folder *steel_belted_radius*.

The sections of the LDAP authentication file are described below.

[Bootstrap] Section

The [Bootstrap] section specifies information that the Steel-Belted Radius/SPE server uses to load and start the LDAP authentication plug-in. You must set the library used, and you must enable LDAP authentication.

This section should look like:

```
[Bootstrap]
LibraryName=ldapauth.so
Enable=1
InitializationString=LDAP
```

[Settings] Section

The [Settings] section forms a basis for all Bind and Search requests against the LDAP server. The information presented here applies to all LDAP servers specified in this file.

Steel-Belted Radius/SPE supports two kinds of LDAP authentication:

- Bind—Steel-Belted Radius/SPE attempts to bind to the LDAP server, using the username and password from the incoming access request (one authentication is performed at one time).
- BindName—Steel-Belted Radius/SPE binds once with credentials to the LDAP server and performs a Search operation against the LDAP server to validate username and password from the incoming access request (multiple authentications are performed at the same time).

The SRC software supports the BindName option, which must be specified in the [Settings] section. The BindName option requires specifying credentials, which Steel-Belted Radius/SPE uses to bind against the LDAP directory. If you want to use the same credentials for each LDAP directory, specify BindName and BindPassword in the [Settings] section; otherwise, use the [Server/name] sections, as described below:

- LogLevel—Activates LDAP logging, written into the activity log file (< *date* > *.log*)
- PasswordFormat—Identifies whether RADIUS handles clear-text, UNIXcrypt, or SHA1 + Base64 hash-encrypted passwords. The value auto instructs Steel-Belted Radius/SPE to parse each password value that it retrieves from the LDAP server.
    - PasswordCase—Tells SPE whether the password is always converted to uppercase or to lowercase or not converted at all. The default is Original.
    - UpperCaseName—Identifies whether the username is converted to uppercase or not. The default is 0 (no conversion).

The **Search** option specifies a string name, referencing to a section where the LDAP Search request is specified.

The section looks like the following:

```
[Settings]
MaxConcurrent=25
Timeout=20
ConnectTimeout=25
QueryTimeout=10
WaitReconnect=2
MaxWaitReconnect=360
LogLevel = 0
UpperCaseName = 0
PasswordCase=original
PasswordFormat=auto
Search = DoLdapSearch
SSL = 0
```

[Server] Section

The [Server] section lists the LDAP servers that may be used to perform authentication. Optionally, it also can be used to specify multiple LDAP servers for load balancing or backup. If more than one LDAP server is specified, Steel-Belted Radius/SPE always uses round-robin. The following depicts how to list one or more LDAP servers.

The list contains serverName = TargetNumber pairs, where the serverName is used in the [server/serverName] section, described in the next paragraph. TargetNumber is an activation target number that controls when the server is activated for backup. TargetNumber is optional and may be left blank. For example:

```
[Server]
s1=
s2=
s3=
```

[Server/serverName] Section

Each [server/serverName] section contains information about a single LDAP server. You must provide a [server/serverName] section for each server you specify in the [server] section. The value for Host identifies the IP address of the LDAP server, and the value for Port specifies the port used for LDAP communications. By default, any LDAP server listens at port 389. The credentials used by Steel-Belted Radius/SPE to bind to the LDAP server are specified in BindName and BindPassword. The SSL value indicates whether an SSL connection is used for the RADIUS-LDAP connection. If the last three mentioned parameters are not specified, Steel-Belted Radius/SPE takes the configuration out of the [Settings] section.

```
[Server/s1]
Host=127.0.0.1
Port=389
BindName=cn=radius,ou=components,o=operators,o=umc
BindPassword=radius
SSL=0

[Server/s2]
```

```
Host=10.20.2.12
Port=389

[Server/s3]
Host=10.10.40.19
Port=389
```

[Search/name] Section

The referenced [Search/name] section includes the search filter, base object, scope, and attribute list, which are included in the LDAP Search operation. If you reference this section in the [Settings] section, the specified options are valid for all LDAP directories. If you want to specify separate Search options for each LDAP directory, you must reference this section in each [server/name] section. In the following example, "DoLdapSearch" is used as name.

NOTE: This name is referenced in the [Settings] section.

Because the SRC software uses the BindName authentication method, you must ensure that the user's password is included in the attribute list, referenced by the attributes option. In the SRC software case, we would like to search only objects where the LDAP attribute uid matches the specified username, and we therefore set *Filter = uid = < User-Name >* . The location within the directory where the search is started is specified in the Base variable. The SRC software for residential users uses the base *retailerName = default, o = Users, o = umc*. The scope of the search is a subtree search (Scope = 2). The variable *%DN* is used for holding the distinguished name of the LDAP search result. The attribute list is a reference to another section of the *ldapauth.aut* file.

```
[Search/DoLdapSearch]
Base=retailerName=default,o=users,o=umc
Scope=2Filter=uid=<User-Name>
Attributes = AttrList
Timeout = 20
%DN = dn
```

For another authentication strategy, see *Configuring Directed Authentication* on page 138. This strategy is more suited for cases in which the service provider outsources services from retailer ISPs.

[Attribute/name] Section

Within the [Attribute/name] section, the LDAP attributes are determined, which are requested by the LDAP search. If the entry that matches the search filter contains values of these attribute types, these values will be part of the search result; RADIUS uses them in the values for checking and replying purposes. Again, the user password attribute is mandatory in the BindName authentication method, which is used in our case. The [Attribute/name] section looks like the following:

```
[Attributes/AttrList]
userPassword
uid
alternateCliAuthLevel
alternateCliVrouterName
```

ascendFilterCmd
atmMBS
atmPCR
atmSCR
atmServiceCategory
cliAllowAllVRAccess
cliInitialAccessLevel
egressPolicyName
egressStatistics
framedIpRouteTag
igmpEnable
ingressPolicyName
ingressStatistics
ipv6LocalInterface
ipv6PrimaryDNS
ipv6SecondaryDNS
ipv6VirtualRouter
localAddressPool
localInterface
pppoeDescription
pppoeMaxSessions
pppoeUrl
qosProfileName
qosProfileInterfaceType
radiusChapPassword
radiusAcctInterimInterval
radiusCalledStationId
radiusCallingStationId
radiusConnectInfo
radiusFilterId
radiusFramedIPAddress
radiusFramedIPNetmask
radiusReplyMessage
radiusFramedProtocol
radiusFramedRoute
radiusFramedPool
radiusSessionTimeOut
radiusNASIdentifier
radiusNASIPAddress
radiusNASPort
radiusNASPortId
radiusNASPortType
radiusClass
radiusIdleTimeOut
radiusServiceType
redirectVRName
pppAuthenticateProtocol
pppPassword
pppUsername
primaryDNS
secondaryDNS
primaryWINS
saValidate
sdxServiceName
sessionVolumeQuota
secondaryWINS
serviceBundle

```
tunnelAssignmentID
tunnelClientEndPoint
tunnelClientAuthID
tunnelMaximumSessions
tunnelMediumType
tunnelNasPortMethod
tunnelPreference
tunnelTOS
tunnelType
tunnelServerEndPoint
tunnelServerAuthID
tunnelPassword
tunnelVirtualRouter
tunnelBearerType
tunnelDialoutNumber
tunnelInterfaceId
tunnelMaximumBps
tunnelMinimumBps
virtualRouterName
```

[Request] Section

In the [Request] section, the incoming RADIUS attributes (from Access-Request) must be determined and mapped to LDAP attributes. Steel-Belted Radius/SPE places these values in the variable table before moving on to the LDAP Bind and Search requests as defined earlier.

```
[Request]
%UserName = User-Name
NAS-IP-Address = radiusNASIPAddress
NAS-Port = radiusNASPort
Service-Type = radiusServiceType
```

[Response] Section

The [Response] section tells Steel-Belted Radius/SPE what to do with the information that it has retrieved from the incoming access request and from the LDAP database. It completes the authentication and issues an access response to the RADIUS client.

```
[Response]
%Password = userpassword
Acct-Interim-Interval = radiusAcctInterimInterval
Address-Pool-Name = localAddressPool
Alt-CLI-Auth-Level = alternateCliAuthLevel
Alt-CLI-Virtual-Router = alternateCliVrouterName
Atm-MBS = atmMBS
Atm-PCR = atmPCR
Atm-SCR = atmSCR
Atm-Service-Category = atmServiceCategory
Class = radiusClass
CLI-Allow-All-VR-Access = cliAllowAllVRAccess
CLI-Initial-Auth-Level = cliInitialAccessLevel
Egress-Policy-Name = egressPolicyName
Egress-Statistics = egressStatistics
Filter-Id = radiusFilterId
```

Framed-IP-Address = radiusFramedIPAddress
Framed-IP-Netmask = radiusFramedIPNetMask
Framed-Ip-Route-Tag = framedIpRouteTag
Framed-Pool = radiusFramedPool
Framed-Route = radiusFramedRoute
Idle-Timeout = radiusIdleTimeOut
Igmp-Enable = igmpEnable
Ingress-Policy-Name = ingressPolicyName
Ingress-Statistics = ingressStatistics
Ipv6-Virtual-Router = ipv6VirtualRouter
Ipv6-Local-Interface = ipv6LocalInterface
Ipv6-Primary-DNS = ipv6PrimaryDNS
Ipv6-Secondary-DNS = ipv6SecondaryDNS
Local-Loopback = localInterface
Ppp-Authenticate-Protocol = pppAuthenticateProtocol
Ppp-Password = pppPassword
Ppp-Username = pppUsername
Pppoe-Max-Sessions = pppoeMaxSessions
Pppoe-Url = pppoeUrl
Primary-DNS = primaryDNS
Primary-WINS = primaryWINS
Qos-Profile-Interface-Type = qosProfileInterfaceType
Qos-Profile-Name = qosProfileName
Redirect-VR-Name = redirectVRName
Sa-Validate = saValidate
Sdx-Service-Name = sdxServiceName
Sdx-Session-Volume-Quota = sessionVolumeQuota
Secondary-DNS = secondaryDNS
Secondary-WINS = secondaryWINS
Service-Type = radiusServiceType
Service-Bundle = serviceBundle
Session-Timeout = radiusSessionTimeOut
Tunnel-Bearer-Type = tunnelBearerType
Tunnel-Dialout-Number = tunnelDialoutNumber
Tunnel-Interface-Id = tunnelInterfaceId
Tunnel-Maximum-Bps = tunnelMaximumBps
Tunnel-Minimum-Bps = tunnelMinimumBps
Tunnel-Assignment-ID = tunnelAssignmentID
Tunnel-Type = tunnelType
Tunnel-Maximum-Sessions = tunnelMaximumSessions
Tunnel-Medium-Type = tunnelMediumType
Tunnel-Nas-Port-Method = tunnelNasPortMethod
Tunnel-Server-Endpoint = tunnelServerEndPoint
Tunnel-Password = tunnelPassword
Tunnel-Preference = tunnelPreference
Tunnel-Tos = tunnelTOS
Tunnel-Virtual-Router = tunnelVirtualRouter
Virtual-Router-Name = virtualRouterName

Configuring Directed Authentication

Directed authentication is used when the service provider manages retailer ISPs. This means that the service provider holds the ISP's end-customer information in its LDAP server, but is not responsible for the data. This data is stored in a separate subtree within the LDAP server.

It is possible that unique identifiers exist in the retailer ISP realm, which might already exist in the service provider realm, or in some other retailer ISP realm. This authentication method allows you to set a different search base, based on the realm name, which is submitted at login time.

Consider an example where the ISP "Virneo" is handled within the service provider's LDAP directory. The service provider and the ISP agreed to use the realm name *virneo.com*.

To configure directed authentication for this example:

1. Enable the realm feature on the RADIUS server (setting parameter in *radius.ini*):

   [Configuration]
   ExtendedProxy = 1

2. Register the realm name with Steel-Belted Radius/SPE (setting parameter in *proxy.ini*):

   [Directed]
   virneo.com

3. Create a realm configuration file called *virneo.com.dir*

NOTE: The filename must be identical to the realm name specified in the previous step.

4. Register the authentication method (LDAP) with the realm (setting parameter in *isp1.com.dir*):

   [AuthMethods]
   VIRNEO.COM

NOTE: The string specified in the [AuthMethods] section must be identical to the LDAP initialization string from the to-be-created authentication file (virneo.com.aut).

5. Enable directed authentication (setting parameter in *virneo.com.dir*), and strip the realm name:

   [Auth]
   Enable = 1
   StripRealm = 1

6. Enable directed accounting (setting parameter in *isp1.net.dir*):

```
[Acct]
Enable = 1
```

7. Define the LDAP configuration interface for directed authentication (creating authentication file *virneo.com.aut*):

   This step is identical to a step mentioned in the *Configuring LDAP Authentication* section. The initialization string in the bootstrap section must be identical to the authentication method, which is specified in *virneo.com.dir*. For example:

```
[Bootstrap]
LibraryName=ldapauth.so
Enable=1
InitializationString=VIRNEO.COM
```

Further details about the proxy configuration and directed realm configurations can be found in the Steel-Belted Radius/SPE manuals.

Customizing the Authentication Log File

The SRC software requires that the RADIUS attribute Class be captured in the accounting files. By default, Steel-Belted Radius/SPE does not include the Class attribute in the accounting files. To accomplish the logging of Class, you must modify the file *account.ini* within the server directory *(/opt/UMC/SPE)* by adding "Class = " to the [Attributes] section:

```
[Attributes]
User-Name=
NAS-Port=
Framed-IP-Address=
Class=
```

Configuring the Steel-Belted Radius/SPE Server and RADIUS Clients

You must configure both the client and server to allow communication between the RADIUS server (SPE 4.0) and the RADIUS clients (the JUNOSe router and the SAE).

Configuring the Steel-Belted Radius Server

The RADIUS server must be able to communicate with the RADIUS clients. The RADIUS server must have the following information for all RADIUS clients connected to the RADIUS server:

- IP address of the RADIUS client
- RADIUS shared secret to be exchanged between the Steel-Belted Radius/SPE server and the client
- Model (vendor) of the RADIUS client

You perform these configurations by using SDX Admin.

Configuring RADIUS Clients

Each RADIUS client must be able to contact its RADIUS server. The RADIUS client must have the following information to allow client/server communication:

- IP address of the RADIUS server
- RADIUS shared secret to be exchanged between the Steel-Belted Radius/SPE server and the client
- UDP ports on which the client sends and receives RADIUS authentication and accounting packets. They must match with the server configuration.

The RADIUS client configuration of the JUNOSe router is described in the *JUNOSe Broadband Access Configuration Guide*.

Using the Radius Administrator to Configure RADIUS Clients

This administration user interface is a Web-based GUI. You use this GUI to configure the JUNOSe router and the SAE as RADIUS clients. Each JUNOSe router and each SAE connected to a Steel-Belted Radius/SPE server must be configured as a RADIUS client.

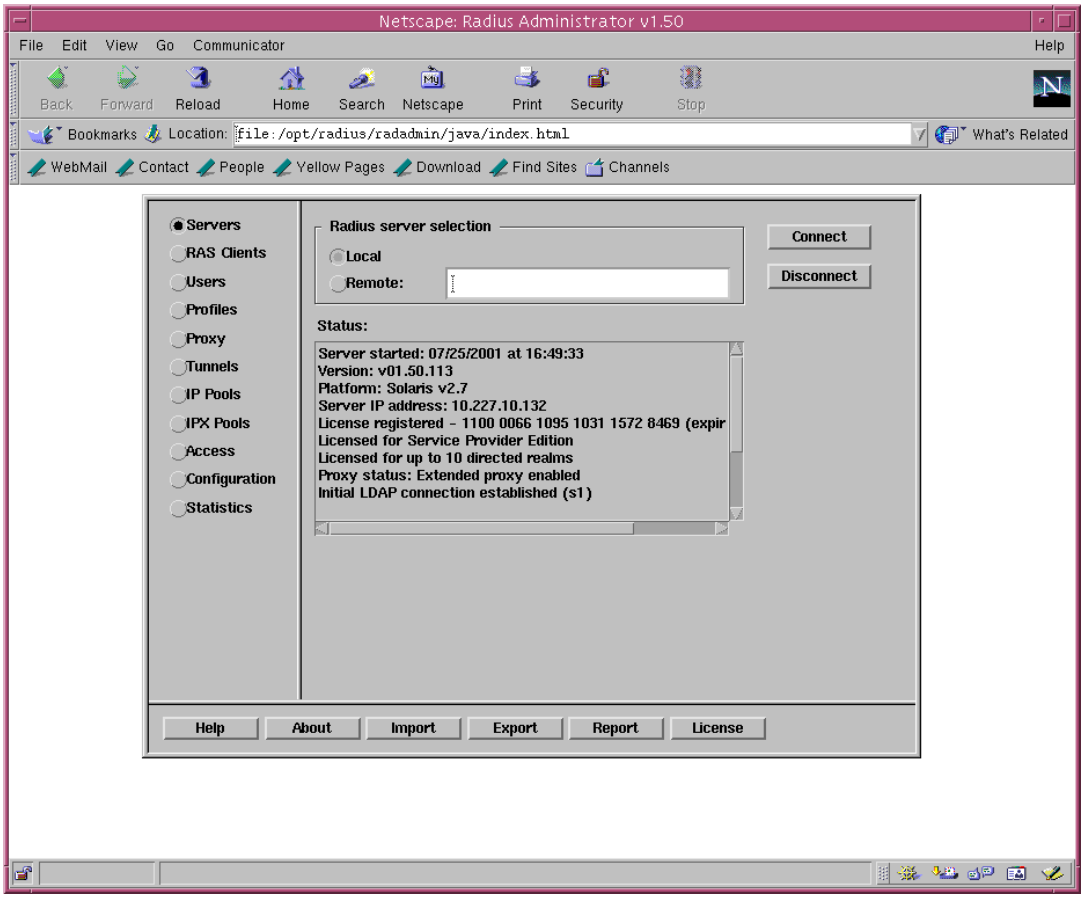
If you have a Netscape browser installed in */opt/netscape*, you must set the Netscape environment variable *MOZILLA_HOME = /opt/netscape* to run Java applets.

After you launch the Netscape browser, choose the URL

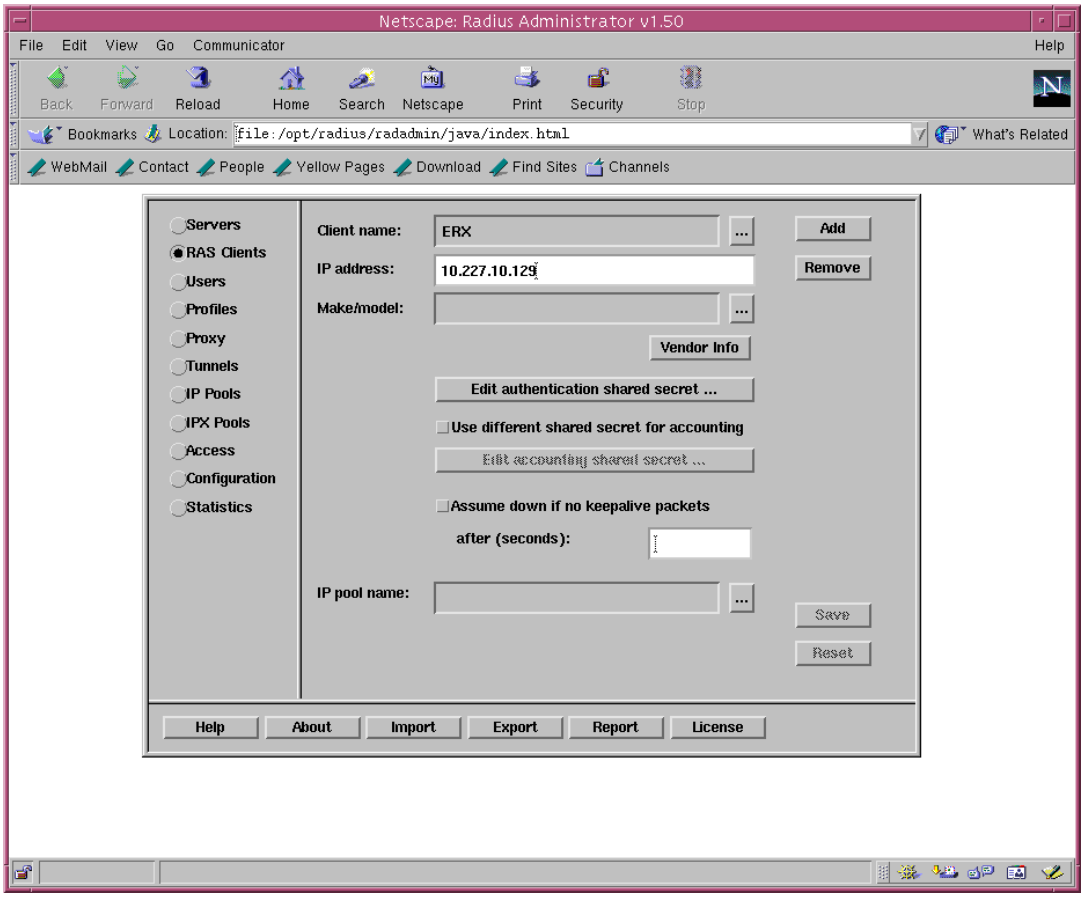
file:///opt/UMC/SPE/radadmin/java/index.html

This file prompts you for authentication credentials when you try to connect to the local server. By default, the username is *admin* with the password *radius*.

After successful authentication, the Web-based GUI is displayed.

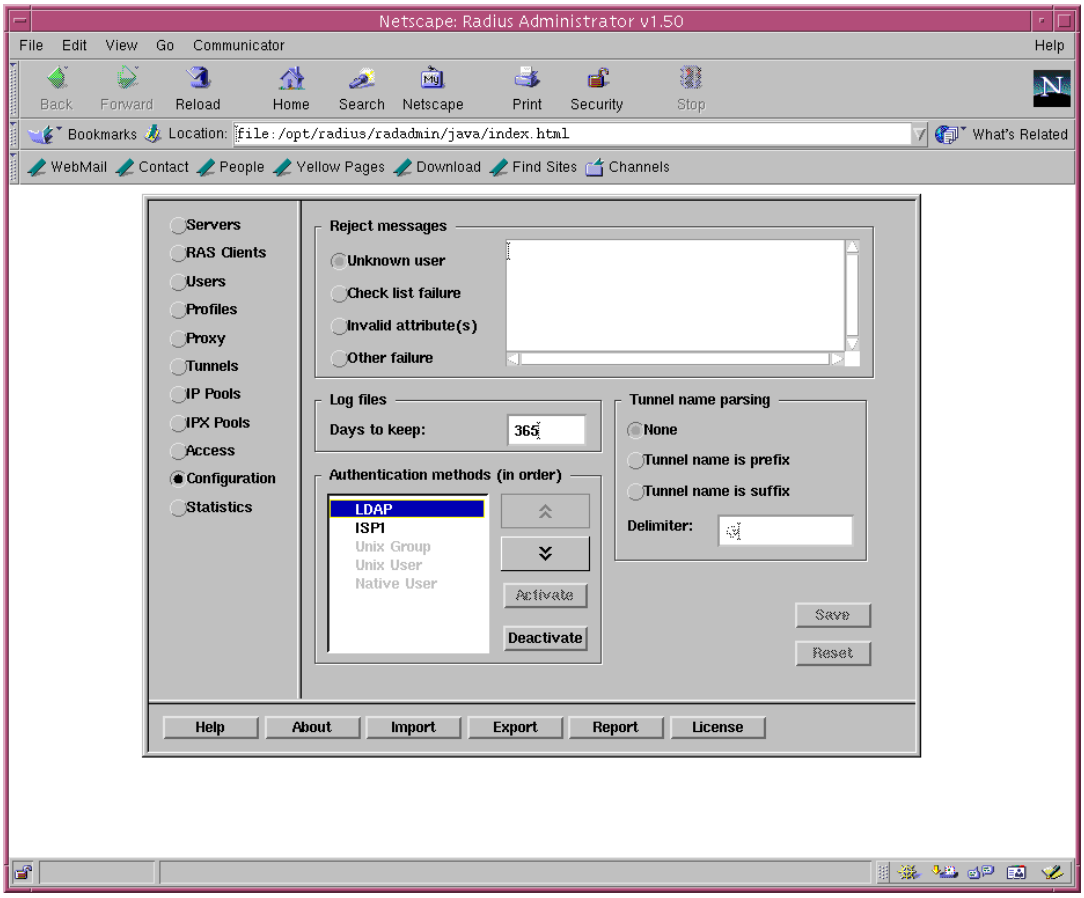


To configure a JUNOSe router as a Remote Access Server (RAS) client, select **RAS Clients** on the left-hand side of the window, and click **Add**; then enter the necessary information.



When the Steel-Belted Radius/SPE server is integrated with the SAE, the Steel-Belted Radius/SPE server must know that the SAE server is a RAS client. The SAE server requires some JUNOSe specific attributes; therefore, it must be configured as a JUNOSe RADIUS client.

To specify the authentication method in the Radius Administration window, select **Configuration** on the left-hand side of the screen. The methods Native User, Unix User, and Unix Group must be deactivated. LDAP will be the only activated method.



Chapter 12

Integrating Merit RADIUS

Use the information in this chapter to integrate Merit AAA with JUNOSe routers. Refer to the *SRC-PE Release Notes* for information about compatibility of this SRC release with Merit AAA 4.22E releases. The SRC software does not support the use of RADIUS with JUNOS routing platforms.

This chapter contains the following sections:

- System Requirements for the Merit AAA Server on page 146
- Installing Merit AAA on page 146
- LDAP Features for the Merit AAA Server on page 146
- Configuring UDP Ports for the Merit AAA Server on page 147
- Starting the Merit AAA Server on page 148
- Stopping the Merit AAA Server on page 149
- Displaying the Status of the Merit AAA Server on page 149
- Extending Dictionary Files with JUNOSe Parameters for the Merit AAA Server on page 149
- Configuring LDAP Authentication for the Merit AAA Server on page 150
- Example: Merit AAA Accounting Log File Format on page 153
- Configuring the Merit AAA Server and RADIUS Clients on page 155
- Testing the Merit AAA Server on page 156

Information about the simpler case of integrating the Merit AAA 4.22E server with the JUNOSe router (without using the SRC software) is provided.

The SRC software can take advantage of a RADIUS server to authenticate against an LDAP server, which is used to store subscriber and service information, among other items.

NOTE: The Merit AAA 4.22E product is provided with the SRC software; all others are third-party products. We recommend that the Merit AAA 4.22E RADIUS solution be used for trial purposes only.

System Requirements for the Merit AAA Server

The following are the system requirements:

- Operating system—Solaris 8 or higher
- RAM—At least 64 MB of working memory
- Disk—Depends on external database support and storage time of the accounting log files; at least 40 MB of hard-disk space

Installing Merit AAA

The Merit AAA 4.22E server package is part of the SRC software distribution and is called UMCradius. The installation procedure for the Merit AAA 4.22E for a Solaris host is described in the *SDX Getting Started Guide, Chapter 28, Installing the SRC Software on a Solaris Platform*.

If the Merit AAA 4.22E server is installed from the SRC software distribution, the dictionary file is already extended by the JUNOSe-specific attributes.

LDAP Features for the Merit AAA Server

The Merit AAA server package is composed of functional building blocks called authentication/authorization transfer vectors (AATVs). These AATVs perform a specific function, such as UNIX password checking or authentication against an LDAP directory.

LDAP authentication allows all user configurations to be done and stored in the LDAP directory, eliminating the need to edit the server's configuration files to change user information. In addition to being a policy repository, the LDAP directory also replaces the user's file or the UNIX password file as the place to store a user ID and password; performance is higher when one is dealing with a large number of users.

The ProLDAP AATV is an authentication AATV that performs two functions. First, it checks the validity of the user's ID and password. Second, if authentication is successful, the AATV loads attribute value pairs into the aaaCheck-list, aaaDeny-list, and aaaReply-list in the authentication request. The ProLDAP AATV uses a set of asynchronous LDAP API functions that allow the ProLDAP to be a direct-type AATV. Using a poll interface inside the AAA server engine, the main process is not blocked by the ProLDAP AATV; therefore, the ProLDAP process does not create and run a child process. The asynchronous LDAP API functions allow an LDAP search, for example, to be sent out to a directory server without waiting for the search result to come back. Later on, the owner of the search may poll the LDAP client to find out if any result is available from the search.

The ProLDAP AATV is designed to work with different LDAP directory configurations. The directory may be configured to either allow or not allow the user password to be returned to the AAA server in an LDAP search. The ProLDAP AATV may be configured to first try searching for the user in the directory. If the password is returned, the ProLDAP AATV makes a password comparison to authenticate the user. Otherwise, the ProLDAP AATV will try to bind the user to the directory with the given password. ProLDAP may be configured to do a bind or search operation, but only if the directories are known to support those configurations.

Configuration of the LDAP search operations based on realms is described in *Configuring LDAP Authentication for the Merit AAA Server* on page 150.

Configuring UDP Ports for the Merit AAA Server

The transaction-based RADIUS protocol uses two UDP ports: one for authentication packets and one for accounting packets. The ports must be configured on two sides: the Merit AAA server and the RADIUS clients (SRC software and JUNOSe router).

The officially assigned UDP port numbers are:

- 1812 for authentication
- 1813 for accounting

Early deployments of RADIUS used 1645/udp for authentication packets and 1646/udp for accounting packets.

The Merit AAA RADIUS server uses the latter ports by default, whereas the JUNOSe router uses the official ports by default.

There are two ways to change these settings:

- Edit the */etc/services* file to contain two entries for RADIUS authentication and accounting service that specify the ports you wish to use:

  radius 1812/udp # RADIUS Authentication
  radacct 1813/udp # RADIUS Accounting

- Override all default and configured values at server startup with the **radiusd -p** and **radiusd -q** command line options. The SRC software installs the Merit AAA server with a start script, called *rad*, which uses ports 1812 and 1813 for authentication and accounting (see next section).

Starting the Merit AAA Server

We include a script for starting the RADIUS server. The filename of the script is *rad*; it is installed in the directory */opt/UMC/radius*.

To start the Merit AAA server:

1. Log in as `root`.
2. Change the directory to */opt/UMC/radius*, and start the program by typing:

```
cd /opt/UMC/radius
./rad start
```

During startup, the RADIUS server binds to the LDAP server. This process requires that the LDAP server be running before the RADIUS server is started.

The RADIUS process is automatically started whenever the Solaris host is started.

If you are using a Merit AAA server that is not supplied by Juniper Networks, you can start the Merit server by launching the RADIUS process.

The syntax is as follows:

```
radiusd -d <conf directory> -da <aaatv directory> -dl <log directory>
-A <acct directory> -n -p <auth port> -q <acct port> -f <fsm file> -pp
<auth relay port> -qq <acct relay port> -g {'syslog' | 'logfile' | 'stderr'} -l
<log format> -t <timeout> -v -z -h
```

where:

- -d—Directory of users, clients, authfile, dictionary, configuration files
- -da—Directory in which the binary AATVs reside
- -dl—Directory into which the log files should go
- -A—Directory in which to put accounting records
- -n—New session table at start for local authorization service (LAS)
- -p—Port number on which to listen for authentication requests
- -q—Port number on which to listen for accounting requests
- -f—Allows the user to specify an alternate finite state machine (FSM) table file instead of the default *radius.fsm* file
- -pp—Port number on which to relay authentication requests

- -qq—Port number on which to relay accounting requests
- -g—Type of logging; select logfile, syslog, or stderr logging
- -t—Inactivity timeout value (minutes)
- -v—Displays RADIUS version
- -h—Displays this help syntax

Stopping the Merit AAA Server

We include a script for stopping the RADIUS server. The filename of the script is *rad*; it is installed in the directory */opt/UMC/radius*.

To stop the RADIUS server:

1. Log in as `root`.
2. Change the directory to */opt/UMC/radius* and stop the program by typing:

   cd /opt/UMC/radius
   ./rad stop

Displaying the Status of the Merit AAA Server

We include a script for displaying the status of the RADIUS server. The filename of the script is *rad*; it is installed in the directory */opt/UMC/radius*.

To check the status of the Merit AAA server:

1. Log in as `root`.
2. Change the directory to */opt/UMC/radius* and display the status by typing:

   cd /opt/UMC/radius
   ./rad status

Extending Dictionary Files with JUNOSe Parameters for the Merit AAA Server

In addition to supporting standard RADIUS attributes, the JUNOSe router supports JUNOSe-specific attributes. These attributes must be introduced to the Merit AAA server. You must use the RADIUS attributes for both Merit AAA server–JUNOSe router integration and Merit AAA server–JUNOSe router–SRC integration. See the *JUNOSe Broadband Access Configuration Guide* for more information about the RADIUS attributes supported by the JUNOSe router.

If you use the Merit AAA server package that we supply, you do not need to extend the dictionary files, and you can proceed to the next section. If, however, you use another version of the Merit AAA server, you must extend the dictionary file.

In such a case, move to the configuration directory of the Merit AAA installation, and edit the dictionary file. Append the JUNOSe-specific attributes to the dictionary file in the following way:

1. Access the directory in which you installed the Merit AAA installation.

   cd /opt/UMC/radius

2. Open the *radius.dct* file.

3. At the end of the file, add the JUNOSe attributes in the ERX RADIUS Dictionary file. To locate the ERX RADIUS Dictionary file, see the JUNOSe software documentation for the supported release on the Juniper Networks Web site at

   http://www.juniper.net/techpubs/software/

The next step defines the JUNOSe router as the network access server (NAS) to be recognized by the Merit AAA server. This involves the extension of the vendor file, which is located in */opt/UMC/radius/etc*.

The vendor file contains a list of zero or more vendor entries. Each vendor entry contains a vendor name and a vendor number. Each entry optionally contains an interim way of mapping external (with respect to the RADIUS server) attribute numbers to internal (with respect to the RADIUS server) vendor-specific attributes. This optional mapping is used on RADIUS requests and responses. The following lines must be added, where every line starting with the character "#" indicates a comment:

```
# Juniper Networks Inc. extensions
ERX-VSA.attr ERX-VSA.value 4874 Juniper
```

Configuring LDAP Authentication for the Merit AAA Server

The SRC software assumes that all RADIUS authentications are performed against the SDX LDAP directory. This section also applies to Merit AAA server integration with a JUNOSe router if Merit AAA authenticates against an LDAP directory. Integration of the JUNOSe-specific attributes, such as primary Domain Name System (DNS), virtual router, and others, must be performed, which is outlined in this section.

Configuring the Merit AAA Server

The Merit AAA server configuration for the ProLDAP AATV is done through the *authfile* file, which is stored in the configuration directory */opt/UMC/radius/etc*. You must configure these tasks:

- How the Merit AAA server performs authentication
- Which external database is used for authentication, based on the realm name

Administrators must create a table in the *authfile* file for each realm name. Merit AAA supports up to four LDAP directories, which could be used for authentication for each realm.

```
realm PROLDAP description
{
Filter-Type bin | cis
Directory directory-1
{
Host dir1.host.com
Port port-number
Administrator directory-manager-dn
[Password directory-manager-password]
SearchBase realm-search-base-in-directory
Authenticate Auto | Bind | Search
}
...
}
```

where

- realm—Identifies the realm name that is used during PPP login (username@realm). The special value NULL specifies treatment of any incoming access request, where no realm name is submitted during the PPP login.
- PROLDAP—Identifies that this table is valid for the ProLDAP AATV.
- Filter-Type—Identifies treatment of the user ID. Valid values are either case sensitive (bin) or not case sensitive (cis).
- Directory—Identifies the start of the directory section. Up to four directory sections are supported per realm. If the value contains spaces or tabs, it must be enclosed by either the double-quote or the single-quote character. Merit AAA uses the round-robin method for those identified directories.
- Host—The value (fully qualified DNS name or IP address) identifies the LDAP directory.
- Port—Identifies the port the LDAP server listens to.
- Administrator—DN that specifies the user entry AAA uses to log in against the LDAP directory. The DN must be specified if Authenticate is set to search.
- SearchBase—DN that represents the start point of the LDAP search operation for that realm.
- Authenticate—Identifies how Merit AAA authenticates incoming access requests. Valid values are:
  - Auto—AAA performs a search as the configured administrator (searches anonymously if no configured administrator), anticipating that the password is in the result. It binds as the user if the password is not available.
  - Bind—AAA tries to bind with the user ID and password specified during the PPP login.

- Search—AAA binds and performs search operation. LDAP returns the user password, which is compared with the password submitted during the PPP login.

NOTE: The SRC software uses the search option.

The following *authfile* example depicts the treatment of PPP logins without any realms and with the realm name *isp1.com*:

```
# This is a realm entry for an LDAP Server with PROLDAP with NO Realm
#
NULL  PROLDAP Default-Setting
{
Filter-Type BIN
Directory SDX
{
Host 123.45.3.1
Port 389
Administrator "cn=radius,ou=components,o=operators,o=umc"
Password      "radius"
SearchBase    "retailerName=default, o=users, o=umc"
Authenticate  search
}
}
# This is a realm entry for two LDAP Server with PROLDAP with Realm isp1.com
#
virneo.com PROLDAP Virneo-Setting
{
Filter-Type BIN
Directory virneo
{
Host 245.3.4.5
Port 389
Administrator "cn=radius,ou=components,o=operators,o=umc"
Password      "radius"
SearchBase    "retailerName=SP,o=users,o=umc"
Authenticate  search
}
    Directory virneo-backup
{
Host 245.3.4.6
Port 389
Administrator "cn=radius,ou=components,o=operators,o=umc"
Password      "radius"
SearchBase    "retailerName=SP,o=users,o=umc"
Authenticate  search
```

After the installation of Merit AAA from the SRC software distribution, the NULL realm is enabled by default.

Configuring RADIUS Profiles with the LDAP Directory

RADIUS servers search objects from the type umcRadiusPerson to authenticate incoming PPP sessions. If RADIUS and JUNOSe-specific attributes must be returned to the JUNOSe router during the authentication process, Merit AAA expects some special AAA attributes:

- aaaReply—A response sent back from the server (for example, a session time limit)
- aaaCheck—An attribute that must be present in the user entry for the entry to evaluate as True
- aaaDeny—An attribute that must NOT be present in the user entry for the entry to evaluate as True

These attributes are multivalued attributes containing the RADIUS attribute value pairs to be processed by the Merit AAA server.

The following depicts a umcRadiusPerson object that returns the RADIUS attribute values for Session-Timeout, Idle-Timeout, and Class, and the JUNOSe-specific attribute for the virtual router to be used on the JUNOSe router. This entry is shown in Lightweight Data Interchange Format (LDIF) notation:

```
dn:serviceName=bras,uniqueID=jane,ou=local,retailerName=isp1,
o=Users,o=umc
objectClass: umcRadiusPerson
objectClass: umcServiceProfile
objectClass: top
uid: jane
userPassword: secret
serviceName: bras1
usedService: serviceName=bras,o=Services,o=umc
aaaReply: Virtual-Router-Name=Default
aaaReply: Class=1,uid,bras
aaaReply: Idle-Timeout=2700
aaaReply: Session-Timeout=10800
```

Example: Merit AAA Accounting Log File Format

The following is an example of an accounting log file generated by Merit AAA with:

- Some accounting activity coming from the JUNOSe RADIUS client (tracking the activity of a PPP session)
- Some accounting activity coming from the SDX RADIUS client (a video service being activated, then deactivated)

NOTE: The Merit AAA server that we supply supports interim accounting by default.

```
Tue May  1 10:58:42 2001
    Acct-Status-Type = Start
    User-Name = "user1@isp1"
    Event-Time = "May  1 2001"
    Acct-Delay-Time = 0
```

```
    NAS-Identifier = "OBIWAN"
    Acct-Session-Id = "erx fastEthernet 3/1::0000022073"
    NAS-IP-Address = 10.227.9.145
    Service-Type = Framed
    Framed-Protocol = PPP
    Framed-IP-Address = 10.227.9.150
    Framed-IP-Netmask = 255.255.255.255
    Framed-Compression = None
    NAS-Port-Type = 15
    NAS-Port = 822083584
    NAS-Port-Id = "fastEthernet 3/1:"
    Ingress-Policy-Name = "unlim"
    Acct-Authentic = RADIUS
    User-Id = "user1"
    User-Realm = "isp1"

Tue May  1 10:59:49 2001
    Acct-Status-Type = Start
    Acct-Delay-Time = 0
    User-Name = "user1@isp1"
    Acct-Session-Id = "sspServiceVideoG:user1:e634da23b6"
    NAS-Identifier = "SSP.lion"
    User-Id = "user1"
    User-Realm = "isp1"

Tue May  1 11:07:25 2001
    Acct-Status-Type = Stop
    Acct-Delay-Time = 0
    User-Name = "user1"
    Acct-Session-Id = "sspServiceVideoG:user1:e634da23b6"
    Acct-Input-Octets = 10681
    Acct-Input-Gigawords = 0
    Acct-Input-Packets = 94
    Acct-Output-Octets = 0
    Acct-Output-Gigawords = 0
    Acct-Output-Packets = 0
    Acct-Session-Time = 456
    NAS-Identifier = "SSP"
    User-Id = "user1"
    User-Realm = ""
    LAS-Start-Time = 988729189
    LAS-Code = LAS-Notlocal
    LAS-Duration = 456
```

Configuring the Merit AAA Server and RADIUS Clients

For the Merit AAA server and RADIUS clients (JUNOSe router and the SAE software) to communicate, you must configure both the client and the server.

Configuring the Merit AAA Server

The RADIUS server must be able to communicate with the RADIUS clients. The following information about all RADIUS clients connected to the RADIUS server must be known to the RADIUS server:

- IP address of the RADIUS client
- RADIUS shared secret to be exchanged between Merit AAA and the client
- Model (vendor) of the RADIUS client

Configure this information by editing the */opt/UMC/radius/etc/clients* file. The client file should look like the following:

```
#Client Name         Key             [type]       [version]        [prefix]
#--------------- -------------- --------------- -------          --------
# SSP Client         192.23.3.10    secret       type=Juniper:NAS    v1
# Juniper ERX node (Enable the Juniper extensions)
                     192.23.3.1     secret       type=Juniper:NAS    v1
```

Configuring RADIUS Clients

Each RADIUS client must be able to contact its RADIUS server. The following information is required for client/server communication:

- IP address of the RADIUS server
- RADIUS shared secret to be exchanged between the Merit AAA server and the client
- UDP ports on which the client sends and receives RADIUS authentication and accounting packets. They must match the server configuration.

The RADIUS client configuration of the JUNOSe router is described in the *JUNOSe Broadband Access Configuration Guide.*

The RADIUS client configuration of the SAE is described in the *SDX Getting Started Guide*.

Testing the Merit AAA Server

The Merit AAA installation from the SRC packages provides a script called *tstrad* for testing the RADIUS setup. This script uses the Merit test tool **radpwtst**. The test script is located in the directory */opt/UMC/radius*. To test the Merit AAA configuration, change to the directory */opt/UMC/radius,* and start the program by typing:

./tstrad <username> <userPassword>

where

<username>—Specifies the string identifying the user during the PPP login; for example, jane@isp1.com

<userPassword>—Specifies the user password submitted during the PPP login; for example, ./tstrad jane@isp1.com secret

If the customer did not install the Merit software from the SRC package, use the **radpwst** tool for testing the Merit AAA configuration by typing:

radpwtst -d <conf directory> -p <auth port> -s <server name> -u <auth type> -x -w < userPassword > <username>

where

- -d—Directory of users, clients, authfile, dictionary, etc.
- -p—Port number on which to listen for authentication requests
- -s—IP address or fully qualified DNS name of the server hosting Merit AAA
- -u—Authentication type; always use ppp
- -x—Allows the user to turn on debugging output
- -w—Allows the user to provide a password on the command line and not be prompted

The following example accomplishes the same as the **tstrad** script:

```
radpwst -d /opt/UMC/radius/etc -p 1812 -s 'hostname' -u ppp -x  -w secret
jane@virneo.com
```

Chapter 13

Integrating RAD-Series RADIUS Server

Use the information in this chapter to integrate the RAD-Series RADIUS Server with JUNOSe routers. See the *SRC-PE Release Notes* for information about compatibility of this SRC release with RAD-Series RADIUS Server releases. The SRC software does not support the use of RADIUS with JUNOS routing platforms.

This chapter contains the following sections:

- System Requirements for the RAD-Series RADIUS Server on page 158
- Installing the RAD-Series RADIUS Server on page 158
- LDAP Features for the RAD-Series RADIUS Server on page 159
- Configuring UDP Ports for the RAD-Series RADIUS Server on page 160
- Starting and Stopping RAD-Series Server Manager on page 161
- Extending Dictionary Files with JUNOSe Parameters for the RAD-Series RADIUS Server on page 163
- Configuring LDAP Authentication for the RAD-Series RADIUS Server on page 163
- Example: RAD-Series RADIUS Server Accounting Log File Format on page 168
- Configuring the RAD-Series RADIUS Server and RADIUS Clients on page 169
- Testing the RAD-Series RADIUS Server on page 170

Information about the simpler case of integrating Interlink Networks RAD-Series RADIUS Server with the JUNOSe router (without using the SRC software) is provided.

The SRC software can take advantage of a RADIUS server to authenticate against an LDAP server, which is used to store subscriber and service information, among other items.

System Requirements for the RAD-Series RADIUS Server

The following system requirements are recommended:

- Operating system—Sun Solaris 8 or Sun Solaris 9
- RAM—At least 128 MB of working memory
- Disk—Depends on external database support and storage time of the accounting log files; at least 50 MB of hard-disk space

Installing the RAD-Series RADIUS Server

You need the RAD-Series RADIUS Server software CD to complete this procedure. You can acquire the software from Interlink Networks, Inc. See

http://www.interlinknetworks.com

To install the RAD-Series RADIUS Server software:

1. Log in as `root`.
2. Change the directory to the location where the installation binary is located. Run the command:

   sh RAD-Series.6.0.solaris.bin

   The system asks for the product features to be installed. Select at least the following features:

   - RADIUS Binary Components
   - RADIUS Configuration Files
   - Server Manager
   - Remote Control

3. Enter the binary directory. For example:

   /opt/UMC/aaa

4. Enter the configuration directory. For example:

   /opt/UMC/aaa/etc

5. When prompted, enter the data directory. For example:

   /opt/UMC/aaa/var

6. Enter the documentation directory. For example:

   /opt/UMC/aaa/doc

7. Enter the path where you want to install Tomcat. For example:

   /opt/UMC/aaa/tomcat

8. When prompted for the shared secret, type:

   secret

9. When prompted for the test user password, type:

   secret

10. When prompted for the Server Manager user, type:

    admin

11. When prompted for the Server Manager password, type:

    radius

    When the installation is complete, the following line appears:

    Installation Complete

    The software has been successfully installed to:

    /opt/UMC/aaa
    /opt/UMC/aaa/etc
    /opt/UMC/aaa/var
    /opt/UMC/aaa/tomcat
    /opt/UMC/aaa/doc

12. To exit the installer, press Enter.

NOTE: See the Interlink Networks RAD-Series RADIUS Server *Getting Started Guide* for information about configuring the server and verifying the installation. The document is located at: */opt/UMC/aaa/doc/doc/gstarted.pdf*.

LDAP Features for the RAD-Series RADIUS Server

The RAD-Series RADIUS Server package is composed of functional building blocks called authentication/authorization transfer vectors (AATVs). These AATVs perform a specific function, such as UNIX password checking or authentication against an LDAP directory.

LDAP authentication allows all user configurations to be done and stored in the LDAP directory, eliminating the need to edit the server's configuration files to change user information. In addition to being a policy repository, the LDAP directory also replaces the user's file or the UNIX password file as the place to store a user ID and password. Performance is higher when one is dealing with a large number of users.

The ProLDAP AATV is an authentication AATV that performs two functions. First, it checks the validity of the user's ID and password. Second, if authentication is successful, the AATV loads attribute value pairs into the aaaCheck-list, aaaDeny-list, and aaaReply-list in the authentication request. The ProLDAP AATV uses a set of asynchronous LDAP API functions that allow an LDAP search, for example, to be sent out to a directory server without waiting for the search result to come back. Later on, the owner of the search may poll the LDAP client to find out if any result is available from the search.

The ProLDAP AATV is designed to work with different LDAP directory configurations. The directory may be configured to either allow or not allow the user password to be returned to the AAA server in an LDAP search. The ProLDAP AATV may be configured to first try searching for the user in the directory. If the password is returned, the ProLDAP AATV makes a password comparison to authenticate the user. Otherwise, the ProLDAP AATV will try to bind the user to the directory with the given password. ProLDAP may be configured to do a bind or search operation, but only if the directories are known to support those configurations.

Configuration of the LDAP search operations based on realms is described in *Configuring LDAP Authentication for the RAD-Series RADIUS Server* on page 163.

Configuring UDP Ports for the RAD-Series RADIUS Server

The transaction-based RADIUS protocol uses two UDP ports: one for authentication packets and one for accounting packets. The ports must be configured on two sides: RAD-Series RADIUS Server and the RADIUS clients (SRC software and JUNOSe router).

The officially assigned UDP port numbers are:

- 1812 for authentication
- 1813 for accounting

Early deployments of RADIUS used 1645/UDP for authentication packets and 1646/UDP for accounting packets.

Both RAD-Series RADIUS Server and the JUNOSe router use the official ports by default. If you decide to use different ports, you can change the port after you start RAD-Series RADIUS Server. See *Starting and Stopping RAD-Series Server Manager* on page 161.

Starting and Stopping RAD-Series Server Manager

To open RAD-Series Server Manager:

1. Start Tomcat by entering:

   /opt/aaa/tomcat/bin/startup.sh

2. Enter the following URL into your Web browser:

   http://<ip-address of server>:8080/aaa/index.html

3. When prompted for the Server Manager username, enter:

   admin

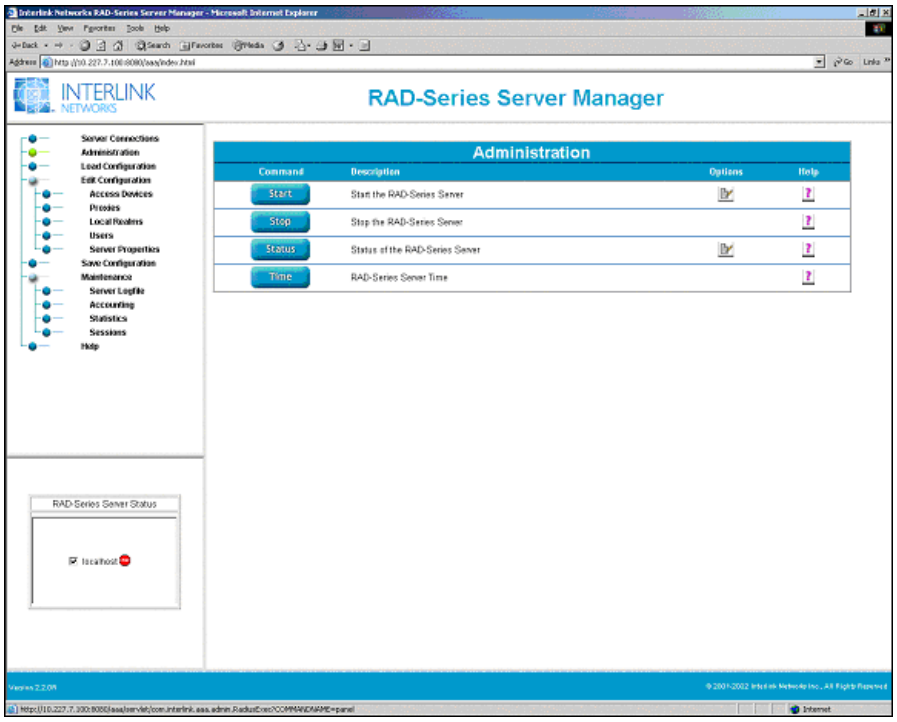
4. When prompted for the Server Manager password, enter:

   radius

NOTE: You must use the same administrator and password that you supplied during the installation.

5. From the navigation pane, click **Administration**.

   The Administration pane appears.



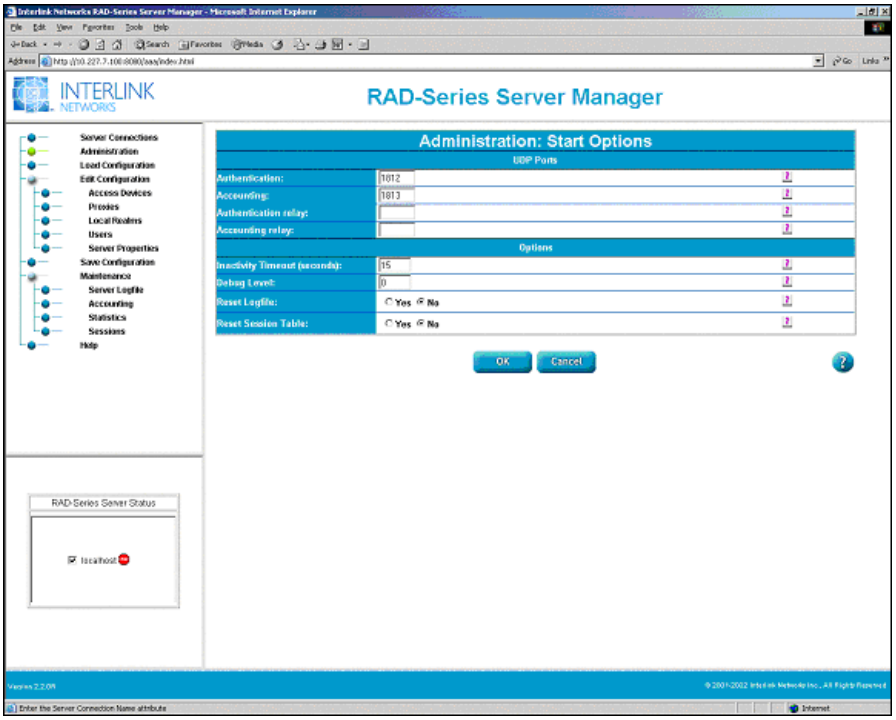
To start or stop RAD-Series Server Manager, click the Start or Stop button. When RAD-Series Server Manager is running, the bullet in the navigation pane turns green. When RAD-Series Server Manager is not running, it is blue.

Changing the UDP Ports

To use UDP ports other than the default ports described in *Configuring UDP Ports for the RAD-Series RADIUS Server* on page 160:

1. Click the **Options** button located to the right of the **Start** button.

   The following pane appears.



2. Under UDP Ports, enter the new Authentication and Accounting port settings.

3. Click **OK**.

Extending Dictionary Files with JUNOSe Parameters for the RAD-Series RADIUS Server

In addition to supporting standard RADIUS attributes, the JUNOSe router supports JUNOSe-specific attributes. These attributes must be introduced to RAD-Series RADIUS Server. You must use the RADIUS attributes for both RAD-Series RADIUS Server–JUNOSe router integration and RAD-Series RADIUS Server–JUNOSe router–SRC integration.

The RAD-Series RADIUS Server package still uses the old Unisphere VSAs in their dictionary file. You must edit the dictionary file (located in */opt/aaa/etc*) and replace the Unisphere attributes with the JUNOSe extensions in the ERX RADIUS Dictionary file. To locate the ERX RADIUS Dictionary file, see the JUNOSe software documentation for the supported release on the Juniper Networks Web site at

http://www.juniper.net/techpubs/software/

The next step defines the JUNOSe router as the network access server (NAS) to be recognized by RAD-Series RADIUS Server. This step involves the extension of the vendor file. The vendor file is located in */opt/aaa/etc*.

The vendor file contains a list of zero or more vendor entries. Each vendor entry contains a vendor name and a vendor number. Each entry optionally contains an interim way of mapping external (with respect to the RADIUS server) attribute numbers to internal (with respect to the RADIUS server) vendor-specific attributes. This optional mapping is used on RADIUS requests and responses. Again, RAD-Series RADIUS Server still uses the Unisphere Networks extension. Edit the vendor file and replace Unisphere with Juniper. The ID should remain at 4874.

The modified lines look like the following:

```
# Juniper Networks
Juniper.attr          Juniper.value         4874    Juniper
```

Configuring LDAP Authentication for the RAD-Series RADIUS Server

The SRC software assumes that all RADIUS authentications are performed against the SDX LDAP directory. This section also applies to RAD-Series Server integration with a JUNOSe router if RAD-Series RADIUS Server authenticates against an LDAP directory.

Configuring the RAD-Series Server Manager

The RAD-Series Server Manager configuration for the ProLDAP AATV is done through the *authfile* file, which is stored in the configuration directory */opt/aaa/etc*. The configuration can be performed either manually by editing the authfile or through the Administration panes of RAD-Series Server Manager. The following methods are to be configured:

- How RAD-Series RADIUS Server authenticates
- Which external database is used for authentication, based on the realm name

Administrators must create a table in the *authfile* file for each realm name.

```
realm PROLDAP description
{
Filter-Type bin | cis

Directory directory-1
{
Host dir1.host.com
Port port-number
Administrator directory-manager-dn
[Password directory-manager-password]
SearchBase realm-search-base-in-directory
Authenticate Auto | Bind | Search
}
...
}
```

where

- realm—Identifies realm name, which is used during PPP login (username@realm). The special value NULL specifies treatment of any incoming access request, where no realm name is submitted during the PPP login.
- PROLDAP—Identifies that this table is valid for the ProLDAP AATV.
- Filter-Type—Identifies the treatment of the user ID. Valid values are either case sensitive (bin) or not case sensitive (cis).
- Directory—Identifies the start of the directory section. Up to four directory sections are supported per realm. If the value contains spaces or tabs, it must be enclosed by either the double-quote or the single-quote character. RAD-Series RADIUS Server uses the round-robin method for those identified directories.
- Host—The value (fully qualified DNS name or IP address) identifies the LDAP directory.
- Port—Identifies the port the LDAP server listens to.
- Administrator—DN, which specifies the user entry that RAD-Series RADIUS Server uses to log in against the LDAP directory. This must be specified if Authenticate is set to Search.
- SearchBase—DN, which represents the starting point of the LDAP search operation for that realm.
- Authenticate—Identifies how RAD-Series RADIUS Server authenticates incoming access requests. Valid values are:
  - Auto—RAD-Series RADIUS Server performs a search as the configured administrator (searches anonymously if no configured administrator), anticipating that the password is in the result. It binds as the user if the password is not available.

- Bind—RAD-Series RADIUS Server tries to bind with the user ID and password specified during the PPP login.
- Search—RAD-Series RADIUS Server binds and performs a search operation. LDAP returns the user password, which is compared with the submitted password during the PPP login.

NOTE: The SRC software uses the search option.

The following *authfile* example depicts the treatment of PPP logins without any realms and with the realm name isp1.com.

```
# This is a realm entry for an LDAP Server with PROLDAP with NO Realm
#
NULL  PROLDAP Default-Setting
{
        Filter-Type BIN
        Directory SSC
        {
                Host 123.45.3.1
                Port 389
                Administrator "cn=umcadmin, o=umc"
                Password      "umc"
                SearchBase    "retailerName=default, o=users, o=umc"
                Authenticate  search
        }
}
# This is a realm entry for two LDAP Server with PROLDAP with Realm isp1.com
#
virneo.com  PROLDAP Virneo-Setting
{
       Filter-Type BIN
       Directory virneo
       {
               Host 245.3.4.5
               Port 389
               Administrator "cn=umcadmin, o=umc"
               Password      "umc"
               SearchBase    "retailerName=SP, o=users, o=umc"
               Authenticate  search
       }
    Directory virneo-backup
       {
               Host 245.3.4.6
               Port 389
               Administrator "cn=umcadmin, o=umc"
               Password      "umc"
               SearchBase    "retailerName=SP, o=users, o=umc"
               Authenticate  search
```

Configuring Realm Administration

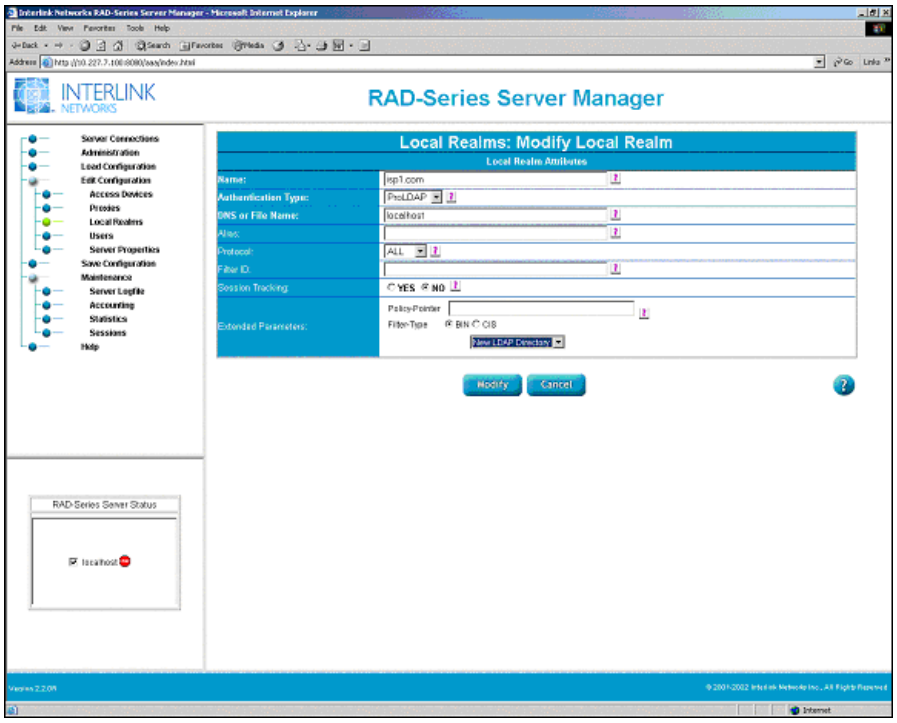
The RAD-Series Server Manager allows you to perform realm administration.

To configure realm administration:

1. From the RAD-Series Server Manager navigation pane, click **Edit Configuration** and **Local Realms**.

2. Click on the **New Local Realm** link.

   The Local Realms: Modify Local Realm pane appears.



3. Specify the realm attributes.

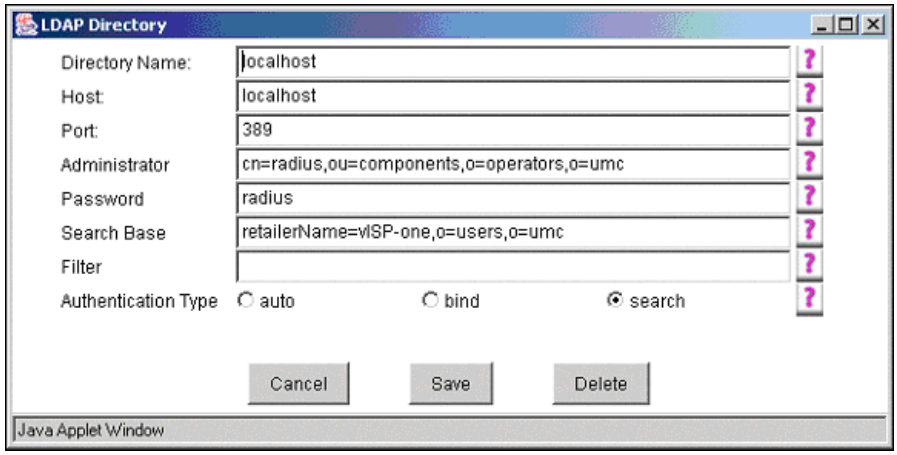
4. Click **Modify**.

Configuring LDAP Settings

To configure the LDAP settings:

1. Select **New LDAP Directory**.

   The LDAP Directory window appears.



2. Specify the attributes.

3. Click **Save**.

Configuring RADIUS Profiles with the LDAP Directory

RADIUS servers search objects from the type umcRadiusPerson to authenticate incoming PPP sessions. If RADIUS and JUNOSe-specific attributes must be returned to the JUNOSe router during the authentication process, RAD-Series RADIUS Server expects some special AAA attributes:

- aaaReply—A response sent back from the server (for example, a session time limit)
- aaaCheck—An attribute that must be present in the user entry for the entry to evaluate as True
- aaaDeny —An attribute that must NOT be present in the user entry for the entry to evaluate as True

These attributes are multivalued attributes containing the RADIUS attribute value pairs to be processed by RAD-Series RADIUS Server.

The following example depicts a umcRadiusPerson object, which returns the RADIUS attribute values for Session-Timeout, Idle-Timeout, and Class, and the JUNOSe-specific attribute for the virtual router to be used on the JUNOSe router. This entry is shown in LDIF notation:

```
dn:serviceName=bras,uniqueID=jane,ou=local,retailerName=isp1,o=Users,
o=umc
objectClass: umcRadiusPerson
objectClass: umcServiceProfile
objectClass: top
uid: jane
userPassword: secret
serviceName: bras1
usedService: serviceName=bras,o=Services,o=umc
aaaReply: Virtual-Router-Name=Default
aaaReply: Class=1,uid,bras
aaaReply: Idle-Timeout=2700
aaaReply: Session-Timeout=10800
```

Example: RAD-Series RADIUS Server Accounting Log File Format

The following is an example of an accounting log file generated by the RAD-Series RADIUS Server with:

- Some accounting activity coming from the JUNOSe RADIUS client (tracking the activity of a PPP session).
- Some accounting activity coming from the SRC RADIUS client (a video service being activated, then deactivated).

```
Tue May  1 10:58:42 2001
    Acct-Status-Type = Start
    User-Name = "user1@isp1"
    Event-Time = "May  1 2001"
    Acct-Delay-Time = 0
    NAS-Identifier = "OBIWAN"
    Acct-Session-Id = "erx fastEthernet 3/1::0000022073"
    NAS-IP-Address = 10.227.9.145
    Service-Type = Framed
    Framed-Protocol = PPP
    Framed-IP-Address = 10.227.9.150
    Framed-IP-Netmask = 255.255.255.255
    Framed-Compression = None
    NAS-Port-Type = 15
    NAS-Port = 822083584
    NAS-Port-Id = "fastEthernet 3/1:"
    Ingress-Policy-Name = "unlim"
    Acct-Authentic = RADIUS
    User-Id = "user1"
    User-Realm = "isp1"

Tue May  1 10:59:49 2001
    Acct-Status-Type = Start
    Acct-Delay-Time = 0
    User-Name = "user1@isp1"
    Acct-Session-Id = "sspServiceVideoG:user1:e634da23b6"
    NAS-Identifier = "SSP.lion"
    User-Id = "user1"
    User-Realm = "isp1"
```

```
Tue May  1 11:07:25 2001
    Acct-Status-Type = Stop
    Acct-Delay-Time = 0
    User-Name = "user1"
    Acct-Session-Id = "sspServiceVideoG:user1:e634da23b6"
    Acct-Input-Octets = 10681
    Acct-Input-Gigawords = 0
    Acct-Input-Packets = 94
    Acct-Output-Octets = 0
    Acct-Output-Gigawords = 0
    Acct-Output-Packets = 0
    Acct-Session-Time = 456
    NAS-Identifier = "SSP"
    User-Id = "user1"
    User-Realm = ""
    LAS-Start-Time = 988729189
    LAS-Code = LAS-Notlocal
    LAS-Duration = 456
```

Configuring the RAD-Series RADIUS Server and RADIUS Clients

For RAD-Series RADIUS Server and RADIUS clients (JUNOSe router and the SAE software) to communicate, you must configure both the client and the server.

Configuring the RAD-Series RADIUS Server

The RADIUS server must be able to communicate with the RADIUS clients. The following information about all RADIUS clients connected to the RADIUS server must be known to the RADIUS server:

- IP address of the RADIUS client
- RADIUS shared secret to be exchanged between RAD-Series RADIUS Server and the client
- Model (vendor) of the RADIUS client

Although the Administration panes allow you to create new clients, we recommend that you edit the */opt/aaa/etc/clients* file when creating new access devices. The client file should resemble the following:

```
#Client Name             Key            [type]     [version]       [prefix]
#----------------  --------------- --------------- ---------       --------
# SAE Client             192.23.3.10    secret     type=Juniper:NAS   v1
# Juniper ERX node (Enable the Juniper extensions)
                         192.23.3.1     secret     type=Juniper:NAS   v1
```

NOTE: The Administration panes do use Juniper in the vendor list. Without changing some HTML files, creating the Juniper RADIUS client will not work when you use the Administration panes.

Configuring RADIUS Clients

Each RADIUS client must be able to contact its RADIUS server. The following information is required for client/server communication:

- IP address of the RADIUS server
- RADIUS shared secret to be exchanged between RAD-Series RADIUS Server and the RADIUS client
- UDP ports on which the RADIUS client sends and receives RADIUS authentication and accounting packets. The ports must match the server configuration.

The RADIUS client configuration of the JUNOSe router is described in the *JUNOSe Broadband Access Configuration Guide*.

Testing the RAD-Series RADIUS Server

You can test the RAD-Series RADIUS Server installation by using the radpwst tool. This tool is located in the */opt/aaa/bin* directory and has the following syntax:

```
radpwtst -d <conf directory>  -p <auth port>  -s <server name> -u <auth type>
-x -w < userPassword > <username>
```

where

- -d—Directory of users, clients, authfile, dictionary, etc. Configuration files
- -p—Port number to listen for auth requests on
- -s—IP Address or fully qualified DNS name of server, hosting RAD-Series RADIUS Server
- -u—Authentication-Type, always use *ppp*
- -x—Allows the user to turn on debugging output
- -w—Allows the user to provide a password on the command line and not be prompted

Include the */opt/aaa/lib* path in your LD_LIBRARY_PATH environment.

You can test your setup by typing:

```
/opt/aaa/bin/radpwst -d /opt/aaa/etc -p 1812 -s 'hostname' -u ppp -x -w secret
jane@virneo.com
```

Index

Index Key

DirX – DirX Directory Server
Merit – Merit AAA 4.22E
OID – Oracle Internet Directory
RAD – RAD-Series AAA RADIUS Server
SPE – Steel-Belted Radius/SPE
Sun – Sun ONE Directory Server

A

aaaCheck
    Merit 145, 151
    RAD 165
aaaDeny
    Merit 145, 151
    RAD 158, 165
aaaReply
    Merit 145, 151
    RAD 158, 165
access control information
    DirX directory server 78
    eTrust Directory 58
    Oracle Internet Directory 64, 66
    Sun ONE Directory Server 71
access control scheme
    activating access
        operator 115
        subscription operator 115
    binding 107
    cachedAuthentication profile 108
    common access rights 116
    directories 106
    directory entries 105
    directory-specific 117
    DirX directory server 117
    lock subtree 111
    mutex group objects 113
    network subtree 112
    parameter subtree 110
    permissions 106
    policy subtree 110
    RADIUS profile 109
    service, policy, and global parameter objects 114
    sspServiceProfile 108
    subscriber, retailer, and service profiles 112
    substitution access 116
    Sun ONE Directory Server 118
    system management 111
    UmcConfiguration 108
    umcRadius Person 109
    umcUser 109
    user class 106
    user subtree 114
    workflow subtree 113
accounting log file
    Merit 151
    RAD 166
    SPE 137
aci attribute (Sun) 71
add-on packages
    DirX directory server 78
    eTrust Directory 57
    Oracle Internet Directory 64
    Sun ONE Directory Server 70–71
administration user interface (SPE) 138
applications
    SRC on CD xiii
assigned IP subscribers
    third-party devices 6, 22
        IP address pools 6, 22
attribute value pairs
    Merit 145, 151
    RAD 158
Attribute/name section, LDAP authentication (SPE) 132
attributes
    AAA
        Merit 151
        RAD 165
    JUNOSe-specific
        integration 129, 148
        Merit 144, 148, 151
        RAD 161, 165, 166
    LDAP 55
Attributes section, customizing authentication file (SPE) 137
audience for documentation xi
authentication
    credentials (SPE) 138
    log file, customizing (SPE) 137

authentication/authorization transfer vectors (AATVs)
Merit ... 144
RAD ... 157
authfile file
Merit ... 148
RAD ... 161

B

backing up
DirX directory ... 84
Sun ONE directory ... 74
Base variable (SPE) ... 132
binary directory (RAD) ... 156
Bind
LDAP authentication (SPE) ... 130
request (SPE) ... 130, 134
binding
access control ... 107
BindName
LDAP authentication (SPE) ... 130
LDAP server (SPE) ... 131
SRC application (SPE) ... 132
user password attribute (SPE) ... 132
BindPassword
LDAP authentication (SPE) ... 130
LDAP server (SPE) ... 131
Bootstrap section, LDAP authentication (SPE) ... 130
browser requirements (SPE) ... 124

C

cacerts files, LDAPS ... 87
cachedAuthentication and access controls ... 108
certificate files, LDAPS ... 87
Class
Merit ... 151
RAD ... 166
SPE ... 137
commands
keytool ... 87
Common Information Model (CIM) ... 49
community manager
configuring, third-party devices
SDX Configuration Editor ... 34
SRC CLI ... 12
configuration
enable LDAP host (SPE) ... 126
files (SPE) ... 125, 126
procedure (SPE) ... 126, 127
configuration directory
Merit ... 148
RAD ... 156
content rules ... 55
conventions defined
icons ... xii
text ... xii
CORBA (Common Object Request Broker Architecture)
reference for SAE
third-party devices ... 31
credentials
authentication (SPE) ... 138
bind to LDAP server (SPE) ... 131
LDAP authentication (SPE) ... 130
crontab file ... 95, 100
customer support ... xvi

D

data directory (RAD) ... 156
data integrators
adding VPNs ... 100
configuring ... 95
deactivating invalid subscriptions to VPNs ... 102
description ... 91
developing ... 94
executing ... 100
getting help ... 93
logging properties ... 95
order of processors ... 96
planning ... 94
processors
Database Reader ... 96
description ... 92
Enterprise Audit File Reader ... 98
LDAP Reader ... 97
LDAP Writer ... 99
suite, installing ... 94
XML File Reader ... 98
XML File Writer ... 99
database
files
installation (SPE) ... 125
previous installations (SPE) ... 126
integrating data into directory ... 92
JDBC drivers ... 94
Database Reader ... 96
dictionary files with JUNOSe parameters
Merit ... 147
RAD ... 161
SPE ... 129
directed accounting (SPE) ... 137
directed authentication (SPE) ... 136
directory
access control scheme ... 105–119
access rights ... 47
provisioning ... 81
reading data ... 92
redundancy ... 46

schema 49–55
supported directory servers 47
transferring data 91
updates to 46
See also LDAP
directory server
See DirX directory server; eTrust Directory; Oracle Internet Directory; Sun ONE Directory Server
DirX directory server
access control scheme 117
add-on package 78
backing up directory 84
directory user 79
dirx user environment 82, 83
installing 80
integrating with SRC software 78–81
loading sample data 81
obtaining 79
restoring directory database 84
standards 78
starting 82, 83
stopping 83
superuser environment 83
uninstalling 82
DirXmetahub 78, 81
DIT (Directory Information Tree)
content rules 55
structure rules 55
dlm1Chassis object class 53
Document Object Model. *See* DOM
document type definitions. *See* DTDs
documentation set, SRC. *See* SRC documentation set
DOM (Document Object Model) 96
DTDs (document type definitions), data integrators . 92

E

Enterprise Audit File Reader 98
Enterprise Service Portal audit plug-in
logs 92
/etc/services file (Merit) 145
eTrust Directory
add-on package 57
installing 59
integrating with SRC software 59
Juniper SDX eTrust Directory
displaying status 61
starting 60
stopping 61
load script 57, 60
loading sample data 60
setup.sh script 58, 59
event notification, third-party devices
configuring properties
SDX Configuration Editor 36
SRC CLI 14
description 7, 23
external database (SPE) 129

G

generate.sh script (DirX) 81
getting help, data integration 93
global parameter objects and access control scheme 114

H

hard-disk space requirements
Merit 144
RAD 156
SPE 124
Host values (SPE) 131

I

icons defined, notice xii
Idle-Timeout 151, 166
initialize.cp flat file (DirX) 78
install.sh script (SPE) 124, 127
installation
procedure (Merit) 144
scripts (SPE) 125, 126
software CD (SPE) 124
installing software
Merit 144
RAD 156
SPE 124
integrating data into directory 48, 92
invalid subscriptions to VPNs 102
IOR
managing SAE
third-party devices 31

J

Java Database Connectivity. *See* JDBC
JDBC (Java Database Connectivity) drivers 94
Juniper Networks dictionary files
Merit 147
RAD 161
SPE 129
Juniper Networks Professional Services 93
Juniper SDX eTrust Directory. *See* eTrust Directory
JUNOSe RADIUS client
Merit 151
RAD 166

K

keytool command 87

L

LDAP
  attributes ... 55
  integration overview ... 45–47
  See also directory; LDAPS
LDAP authentication
  AATV
    Merit ... 144
    RAD ... 157
  Attribute/name section (SPE) ... 132
  Bind (SPE) ... 130
  bind credentials (SPE) ... 130
  BindName (SPE) ... 130
  Bootstrap section (SPE) ... 130
  configuring
    Merit ... 148
    SPE ... 129
  enabling
    SPE ... 126
  file, sections (SPE) ... 129
  Request section (SPE) ... 134
  Response section (SPE) ... 134
  Search/name section (SPE) ... 132
  Server section (SPE) ... 131
  server/serverName section (SPE) ... 131
  Settings section (SPE) ... 130
LDAP configuration
  interface (SPE) ... 137
  RAD ... 165
LDAP database (SPE) ... 134
LDAP directory
  Merit ... 144, 148
  SPE ... 136
LDAP features
  Merit ... 144
  RAD ... 157
LDAP host (SPE) ... 126
LDAP libraries (SPE) ... 127
LDAP Reader ... 97
LDAP server
  Merit ... 146
  SPE ... 128, 130
LDAP Writer ... 99
ldapauth.aut file (SPE) ... 129
LDAPS
  authentication and connection sequence ... 85
  cacerts files ... 87
  certificate files ... 87
  configuring
    directory server ... 86
    overview ... 86
    SAE components ... 88, 89
  directory connections ... 48
  disabling connection for SAE components ... 89
  enabling connection for SAE components ... 88
  overview ... 85
LDIF (LDAP Data Interchange Format) files ... 48, 56
license string
  SPE ... 124, 126
Lightweight Data Interchange Format (LDIF) (Merit) ... 151
load balancing (SPE) ... 131
lock subtree and access control scheme ... 111
log.txt log file (DirX) ... 81
logging
  properties
    data integrators ... 95
login process
  assigned IP subscribers, third-party devices ... 6, 22
  event notification method, third-party devices ... 7, 23
LogLevel (SPE) ... 130
logs, Enterprise Service Portal audit plug-in ... 92

M

manuals, SRC
  comments ... xv
Merit AAA server
  Oracle Internet Directory integration, with ... 63
Merit server
  command syntax ... 146, 154
  testing ... 154
meta agent ... 81
metacontroller ... 81
metadirectory store ... 81
mutex group
  access control scheme ... 113

N

naming syntax ... 48
network and access control scheme ... 112
notice icons defined ... xii

O

object classes ... 51–54
operating system requirements
  Merit ... 144
  RAD ... 156
  SPE ... 124
operators
  activating access ... 115
Oracle Internet Directory
  add-on package ... 64
  directory settings ... 66
  installing ... 65
  integrating with SRC software ... 64–66
  load script ... 64, 66
  loading sample data ... 66

Merit AAA server, and 63
password configuration 64
order, processes in data integrators 96
outsourced services
retailer ISP, strategy (SPE) 132

P

parameters
subtree and access control scheme 110
password, authentication credentials (SPE) 138
PasswordCase (SPE) 130
PasswordFormat (SPE) 130
PCIM (Policy Core Information Model) 53
permissions in access control lists 106
planning
data integration 94
policy objects
access control 114
policy subtree and access control scheme 110
ports
UDP
Merit 145
RAD 158
SPE 128
processors for data integration 92
ProLDAP AATV
LDAP directory (Merit) 145
Merit 145
RAD 158, 161, 162
server configuration (Merit) 148
ProLDAP process (Merit) 145
properties
processors in data integrators 95
property files
data integrators 95

R

RADIUS
first-time installations (SPE) 124
previous installations (SPE) 125
profiles and access control scheme 109
RADIUS accounting
Merit 145
RADIUS attributes
dictionary files
Merit 147
RAD 161
SPE 129
incoming from Access-Request (SPE) 134
values, Merit 151
RADIUS authentication
Merit 145
RAD 161
SPE 129
RADIUS client
configuration
Merit 153
RAD 168
SPE 138
UDP ports
Merit 145
RAD 158
SPE 127
vendor
Merit 153
RAD 167
RADIUS process 124
Merit 146
SPE 125, 128
RADIUS profiles
configuring (Merit) 151
Merit 151
RAD 165
RADIUS protocol, transaction-based (SPE) 127
RADIUS server
configuring
Merit 153
RAD 167
SPE 137
displaying status
Merit 147
starting
Merit 146
RAD 159
SPE 128
stopping
Merit 147
RAD 159
SPE 128
testing installation (RAD) 168
RADIUS shared secret
Merit 153
RAD 167, 168
radius.ini file (SPE) 128
RAD-Series RADIUS Server
installation 156
syntax description 168
testing 168
RAD-Series Server Manager 159
RAM requirements
Merit 144
RAD 156
SPE 124
RAS client, configuring (SPE) 140
realm
administration (RAD) 164
configuration file (SPE) 136
syntax (Merit) 149

realm name
authentication (RAD) ... 161
description (RAD) ... 162
directed authentication (SPE) ... 136
release notes ... xv
Request section, LDAP authentication (SPE) ... 134
Response section, LDAP authentication (SPE) ... 134
restoring
DirX directory database ... 84
Sun ONE directory database ... 75
retailer ISP
directed authentication ... 136
realm name (SPE) ... 136
retailers
access control ... 112
round-robin method
Merit ... 149
SPE ... 131
router object
adding for third-party devices
SDX Admin ... 26
SRC CLI ... 10

S

SAE (service activation engine)
security properties ... 88
SAE (service activation engine), configuring
IOR, third-party devices ... 31
SAE communities
configuring, third-party devices
Solaris platforms ... 32
SRC CLI ... 12
description, third-party devices ... 4, 20
sample data ... 56
loading
DirX directory server ... 81
eTrust Directory ... 60
Oracle Internet Directory ... 66
Sun ONE Directory Server ... 73
sample data integrators ... 100
schema, loading
DirX directory server ... 78
Oracle Internet Directory ... 64
Sun ONE Directory Server ... 71
script services
for third-party devices ... 5, 21
scripts
generate (DirX) ... 78
install.sh (SPE) ... 125
load (eTrust Directory) ... 57
load (OID) ... 64, 66
load (Sun) ... 73
setup.sh (eTrust Directory) ... 58
vpndatamgt ... 95
SDX RADIUS client (Merit) ... 151
Search requests (SPE) ... 130, 134
Search/name section, LDAP authentication (SPE) ... 132
Server Manager
password (RAD) ... 157
user (RAD) ... 157
Server section, LDAP authentication (SPE) ... 131
Server/name, LDAP authentication (SPE) ... 130
server/serverName section, LDAP authentication (SPE) ... 131
service objects, access control ... 114
service providers
access control ... 112
session store
in third-party networks ... 4, 20
Session-Timeout
Merit ... 151
RAD ... 166
Settings section, LDAP authentication (SPE) ... 130
shared secret (RAD) ... 157
silent installation file (Sun) ... 70
SNMP
retrieving information from network devices 18, 41
software
CD installation (SPE) ... 124
SQL queries ... 94
SRC documentation set
comments ... xv
obtaining ... xv
SRC documentation CD ... xiii
SRC software distribution ... xv
SSL (secure sockets layer). *See* LDAPS
sspServiceProfile and access control scheme ... 108
Steel-Belted Radius/SPE, software requirements ... 124
structure rules ... 55
subscribers
access control scheme ... 112
subscriptions
activating access for operators ... 115
substitutions
access ... 116
Sun ONE Directory Server
access control ... 118
add-on package ... 70–71
backing up directory ... 74
installing ... 72
integrating with SRC software ... 71–73
load script ... 73
loading sample data ... 73
obtaining ... 71
product description ... 69
restarting ... 74
restoring directory ... 75
silent installation feature ... 70

starting 73
stopping 74
superuser environment (DirX) 83
support, requesting xvi
syntax for directory entries 48
system management
access control scheme 111
system requirements
Merit 144
RAD 156
SPE 124

T

TargetNumber (SPE) 131
technical support, requesting xvi
test script (Merit) 154
test user password (RAD) 157
text conventions defined xii
third-party devices
creating sessions 5, 21
initialization scripts 15, 37
integrating into SRC network
Solaris platform 19–42
SRC CLI 3–18
logging in subscribers
assigned IP method 6, 22
event notification method 7, 23
overview 5, 21
provisioning with script services 5, 21
router objects, adding
SDX Admin 26
SRC CLI 10
SAE communities 4, 20
VR objects, adding
SDX Admin 28
SRC CLI 11
Tomcat installation (RAD) 156
transaction-based RADIUS protocol
Merit 145
RAD 158
SPE 127
transferring
data to directory 91

U

UDP ports
changing default
Merit 145
RAD 158, 160
Merit 145
RAD 158, 168
RADIUS client
Merit 153
SPE 138
SPE 127, 128
uid-uniqueness plug-in (Sun) 70
UmcConfiguration and access controls 108
umcRadius Person and access control scheme 109
umcUser and access control scheme 109
Unisphere VSA, updating (RAD) 161
UNIX password checking (Merit) 144
UpperCaseName (SPE) 130
user class in access control lists 106
user password, specifying (Merit) 154
user subtree and access control scheme 114
username, authentication credentials (SPE) 138

V

variables.tcl file (DirX) 78
vendor files
Merit 148
RAD 161
SPE 127
vendor-specific attributes
Merit 148
RAD 161
virtual routers
adding for third-party devices
SDX Admin 28
SRC CLI 11
VPN Subscription Deactivator 102
vpndatamgt script 95
VPNs (virtual private networks)
adding
data integrator 100

W

Workflow application
elements in directory 54
workflow subtree and access control scheme 113

X

XML documents 92
XML File Reader 98
XML File Writer 99
XSLT files 94
XSLT transformers 92