



SRC-PE Software

Integration Guide: Network Devices, Directories, and RADIUS Servers

Release 2.0.x

Juniper Networks, Inc.

1194 North Mathilda Avenue
Sunnyvale, CA 94089

USA

408-745-2000

www.juniper.net

Juniper Networks, the Juniper Networks logo, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. JUNOS and JUNOSe are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

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

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

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

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

Products made or sold by Juniper Networks (including the ERX-310, ERX-705, ERX-710, ERX-1410, ERX-1440, M5, M7i, M10, M10i, M20, M40, M40e, M160, M320, and T320 routers, T640 routing node, and the JUNOS, JUNOSe, and SDX-300 software) or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Copyright © 2007, Juniper Networks, Inc.
All rights reserved. Printed in USA.

SRC-PE Software Integration Guide: Network Devices, Directories, and RADIUS Servers, Release 2.0.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
11 October 2007—Revision 1

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

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 NON-INFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

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

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

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

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

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

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

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

Table of Contents

About This Guide	ix
Objectives	ix
Audience	ix
Documentation Conventions	x
Related Juniper Networks Documentation	xi
Obtaining Documentation	xiii
Documentation Feedback	xiii
Requesting Support	xiv

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 Controller	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

Part 2**Integrating Directories**

Chapter 2	Overview of LDAP Integration	21
	LDAP Overview	21
	Directory Availability	22
	Directory Updates	22
	Supported Directories	23
	Directory Security	23
	Directory Access	23
	LDAPS Directory Connections	24
	Provisioning the Directory	24
	Naming Directory Entries	24
	SDX Directory Schema and Object Model	25
	Naming Convention for Entries	26
	Directory Schema for SRC Software	27
	Object Classes	27
	Objects Representing Folders	27
	Subscriber Objects	28
	Service Objects	29
	Subscription Profile Objects	29
	Policy Objects	29
	Network Device Objects	29
	Configuration and System Management	30
	Attributes	30
	Structure Rules	31
	Content Rules	31
	Where to Find More Information About the Object Model and Directory Schema	32
Chapter 3	Integrating eTrust Directory	33
	Overview of Integration with eTrust Directory	33
	About the eTrust Directory Add-On Package	33
	Integrating the eTrust Directory with the SRC Software	35
	Installing eTrust Directory to Integrate with the SRC Software	35
	Configuring the SDX eTrust Directory Server Agent with the SRC Software	35
	Starting SDX eTrust Directory	36
	Stopping SDX eTrust Directory	37
	Displaying the Status of SDX eTrust Directory	37
	Backing Up and Restoring eTrust Directory	37
Chapter 4	Integrating Oracle Internet Directory	39
	Overview of Oracle Internet Directory Integration	39
	About the Oracle Internet Directory Add-On Package	40
	Integrating the Oracle Internet Directory with the SRC Software	40
	Installing Oracle Internet Directory to Integrate with the SRC Software	41
	Before You Install Oracle Internet Directory	41
	Specifying Configuration Values During Installation	41
	Verifying Directory Settings	42
	Running the Load Script for the Oracle Internet Directory Integration	42
	Starting and Stopping Oracle Internet Directory	42

	Setting Up Local Configuration for SRC Components	43
	Backing Up and Restoring the Oracle Internet Directory	43
Chapter 5	Integrating Sun ONE Directory Server	45
	Overview of Sun ONE Directory Server Integration	45
	About the Sun ONE Add-On Package	46
	Silent Installation for Sun ONE Directory Server	46
	Load Script to Integrate Sun ONE Directory Server	47
	Integrating the Sun ONE Directory with the SRC Software	47
	Installing the Sun ONE Directory Add-On Package	48
	Configuring an Instance of Sun ONE Directory Server	48
	Starting Sun ONE Directory Server	49
	Stopping Sun ONE Directory Server	50
	Restarting Sun ONE Directory Server	50
	Backing Up the Sun ONE Database	50
	Restoring the Sun ONE Database	51
Chapter 6	Integrating the DirX Directory Server	53
	Overview of DirX Directory Server Integration	53
	About the DirX Add-On Package	54
	Integrating the DirX Directory with the SRC Software	54
	Preparing to Install the DirX Directory Server	55
	Installing the DirX Directory Server	56
	Installing the UMCdirxa Add-On Package	56
	Configuring the DirX Directory Server	56
	Provisioning the Directory by Using DirXmetahub	57
	Uninstalling the DirX Directory Server	58
	Starting the DirX Directory Server	58
	Starting the DirX Directory Server in a dirx user Environment	58
	Starting the DirX Directory Server in a Superuser Environment	59
	Stopping the DirX Directory Server	59
	Stopping the DirX Directory Server in a dirx user Environment	59
	Stopping the DirX Directory Server in a Superuser Environment	59
	Backing Up the DirX Database	60
	Restoring the DirX Directory Database	60
Chapter 7	Configuring LDAPS for SRC Components	61
	Overview of LDAPS Support	61
	LDAPS Authentication and Connection	61
	Configuring LDAPS Connections	62
	Configuring the Directory Server to Support LDAPS Connections	62
	Establishing Trust for Directory Clients	63
	Configuring the SAE to Find the Certificate Store	63
	Enabling LDAPS Communication for SAE Components	64
	Disabling LDAPS Communication for SAE Components	65
Chapter 8	Access Control Scheme	67
	Directory Configuration	67
	Directories	68
	User Class	68
	Permissions	68
	Access Controls	69

Access Controls for the Entire Tree	69
Access Controls Against Objects from Type cachedAuthentication Profile and UmcConfiguration	70
Access Controls Against sspServiceProfile	70
Access Controls Against umcRadius Person and umcUser	71
Access Controls Against RADIUS Profiles	71
Access Controls Against the Policy Subtree	72
Access Controls Against the Parameter Subtree	72
Access Controls for System Management	73
Access Controls Against the Lock Subtree	73
Access Controls Against the Network Subtree	74
Access Controls Against Services and Mutex Group Objects	74
Access Controls Against the User Subtree	75
Access Controls Against Service, Policy, and Global Parameter Objects...	75
Activation Access Rights	76
Subscription Access Rights	76
Substitution Access Rights	77
Common Access Rights for All Managers	77
Directory-Specific Access Control Implementation	78
DirX Directory Server	78
Sun ONE Directory Server	79

Part 3

Integrating RADIUS Servers

Chapter 9	Integrating Steel-Belted Radius/SPE Server	83
	System Requirements for the Steel-Belted Radius Server	83
	Installing the Steel-Belted Radius/SPE Software	83
Chapter 10	Integrating RAD-Series RADIUS Server	85
	System Requirements for the RAD-Series RADIUS Server	86
	Installing the RAD-Series RADIUS Server	86
	LDAP Features for the RAD-Series RADIUS Server	87
	Configuring UDP Ports for the RAD-Series RADIUS Server	88
	Starting and Stopping RAD-Series Server Manager	89
	Changing the UDP Ports	90
	Extending Dictionary Files with JUNOS Parameters for the RAD-Series RADIUS Server	91
	Configuring LDAP Authentication for the RAD-Series RADIUS Server	91
	Configuring the RAD-Series Server Manager	92
	Configuring Realm Administration	94
	Configuring LDAP Settings	95
	Configuring RADIUS Profiles with the LDAP Directory	95
	Example: RAD-Series RADIUS Server Accounting Log File Format	96
	Configuring the RAD-Series RADIUS Server and RADIUS Clients	97
	Configuring the RAD-Series RADIUS Server	97
	Configuring RADIUS Clients	98
	Testing the RAD-Series RADIUS Server	98
	Index	99

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 ix](#)
- [Audience on page ix](#)
- [Documentation Conventions on page x](#)
- [Related Juniper Networks Documentation on page xi](#)
- [Obtaining Documentation on page xiii](#)
- [Documentation Feedback on page xiii](#)
- [Requesting Support on page xiv](#)

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 JUNOS routers and JUNOS routing platforms in an Internet access environment. We assume that readers know how to use the routing platforms, directories, and RADIUS servers that they will deploy in their SRC networks. 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	<ul style="list-style-type: none"> Represents keywords, scripts, and tools in text. Represents a GUI element that the user selects, clicks, checks, or clears. 	<ul style="list-style-type: none"> Specify the keyword exp-msg. Run the install.sh script. Use the pkgadd tool. To cancel the configuration, click Cancel.
Bold sans serif typeface	Represents text that the user must type.	<code>user@host# set cache-entry-age cache-entry-age</code>
Monospace sans serif typeface	Represents information as displayed on your terminal's screen, such as CLI commands in output displays.	<pre> nic-locators { login { resolution { resolver-name /realms/login/A1; key-type LoginName; value-type SaeId; } } } </pre>

Table 2: Text Conventions (continued)

Convention	Description	Examples
Regular sans serif typeface	<ul style="list-style-type: none"> ■ Represents configuration statements. ■ Indicates SRC CLI commands and options in text. ■ Represents examples in procedures. ■ Represents URLs. 	<ul style="list-style-type: none"> ■ <code>system ldap server { stand-alone;</code> ■ Use the <code>request sae modify device failover</code> command with the <code>force</code> option. ■ <code>user@host# . . .</code> ■ <code>http://www.juniper.net/techpubs/software/management/sdx/api-index.html</code>
<i>Italic sans serif typeface</i>	Represents variables in SRC CLI commands.	<code>user@host# set local-address local-address</code>
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <code><gfwif></code> .
Key name	Indicates the name of a key on the keyboard.	Press Enter.
Key names linked with a plus sign (+) .	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
<i>Italic typeface</i>	<ul style="list-style-type: none"> ■ Emphasizes words. ■ Identifies chapter, appendix, and book names. ■ Identifies distinguished names. ■ Identifies files, directories, and paths in text but not in command examples. 	<ul style="list-style-type: none"> ■ There are two levels of access: <i>user</i> and <i>privileged</i>. ■ <i>Chapter 2, Services</i>. ■ <i>o = Users, o = UMC</i> ■ The <i>/etc/default.properties</i> file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	<code>Plugin.radiusAcct-1.class = \net.juniper.smgmt.sae.plugin\RadiusTrackingPluginEvent</code>
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.)	<code>diagnostic line</code>

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*.

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	
<i>C2000 and C4000 Hardware Guide</i>	Describes the hardware platforms and how to install, maintain, replace, and troubleshoot them. The guide also includes specifications.
<i>C2000 and C4000 Quick Start Guide</i>	Describes how to get the C-series Controller up and running quickly. Intended for experienced installers who want to expedite the installation process.
<i>SRC-PE Getting Started Guide</i>	Describes the SRC software, how to set up an initial software configuration, and how to upgrade the SRC software. It also explains how to manage a C-series Controller. The guide describes how to set up and start the SRC CLI and the C-Web interface, as well as other SRC configuration tools. It includes reference material for the SRC documentation.
<i>SRC-PE CLI User Guide</i>	Describes how to use the SRC CLI, configure and monitor the platform with the CLI, and control the CLI environment. The guide also describes how to manage SRC components with the CLI.
<i>SRC-PE Network Guide: SAE, Juniper Networks Routers, NIC, and SRC-ACP</i>	Describes how to use and configure the SAE, the NIC, and the SRC-ACP (Admission Control Plug-In) application. This guide also provides detailed information for using JUNOS routers and JUNOS routing platforms in the SRC network.
<i>SRC-PE Integration Guide: Network Devices, Directories, and RADIUS Servers</i>	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.
<i>SRC-PE Services and Policies Guide</i>	Describes how to work with services and policies. The guide provides an overview, configuration procedures, and management information. The guide also provides information about the SRC tools for configuring policies.
<i>SRC-PE Subscribers and Subscriptions Guide</i>	Describes how to work with residential and enterprise subscribers and subscriptions. The guide provides an overview, configuration procedures, and management information. This guide also provides information about the enterprise service portals, including the Enterprise Manager Portal.
<i>SRC-PE Monitoring and Troubleshooting Guide</i>	Describes how to use logging, the SNMP agent, the SRC CLI, and the C-Web interface to monitor and troubleshoot SRC components. This guide also describes the SNMP traps.
<i>SRC-PE Solutions Guide</i>	Provides high-level instructions for SRC implementations. The guide documents the following scenarios: managing QoS services on JUNOS routers; managing subscribers in a wireless roaming environment; providing voice over IP (VoIP) services; integrating the SRC software in a PCMM environment, including the use of the Juniper Policy Server (JPS); and mirroring subscriber traffic on JUNOS routers.
<i>SRC-PE CLI Command Reference, Volume 1</i> <i>SRC-PE CLI Command Reference, Volume 2</i>	Together constitute information about command and statement syntax; descriptions of commands, configuration statements, and options; editing level of statement options; and a history of when a command was added to the documentation.
<i>SRC-PE NETCONF API Guide</i>	Describes how to use the NETCONF application programming interface (API) to configure or request information from the NETCONF server on a C-series Controller that runs the SRC software.
<i>SRC-PE XML API Configuration Reference</i>	Describes the tag elements in the SRC Extensible Markup Language (XML) application programming interface (API) that are equivalent to configuration statements in the SRC command-line interface (SRC CLI).
<i>SRC-PE XML API Operational Reference</i>	Describes the tag elements in the SRC Extensible Markup Language (XML) application programming interface (API) that are equivalent to operational commands in the SRC command-line interface (SRC CLI).

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

Document	Description
<i>SRC-PE Comprehensive Index</i>	Provides a complete index of the SRC guides, excluding the <i>C-series Hardware Guide</i> , the <i>SRC CLI Command Reference</i> , the <i>SRC-PE NETCONF API Guide</i> , the <i>SRC-PE XML API Configuration Reference</i> , and the <i>SRC-PE XML API Operational Reference</i> .
Application Library	
<i>SRC Application Library Guide</i>	Describes how to install and work with applications that you can use to extend the capabilities of the SRC software. The guide documents the following applications: SRC-SG (SOAP Gateway) Web applications, an application to provide threat mitigation, an application to provide tracking and QoS control at the application level by integrating the SRC software with the Ellacoya deep packet inspection (DPI) platform, and an application to control volume usage.
Release Notes	
<i>SRC-PE Release Notes</i>	In the <i>Release Notes</i> , you will find the latest information about features, changes, known problems, resolved problems, supported platforms and network devices (such as Juniper Networks routers and CMTS devices), and third-party software. If the information in the <i>Release Notes</i> differs from the information found in the documentation set, follow the <i>Release Notes</i> . Release notes are included in the corresponding software distribution and are available on the Web.
<i>SRC Application Library Release Notes</i>	

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 Controller.

You can also use the C-Web interface to integrate third-party devices. For more information, see *SRC-PE C-Web Interface Configuration Guide, Chapter 20, Integrating Third-Party Network Devices into the SRC Network with the C-Web Interface*.

Topics in this chapter include:

- [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.

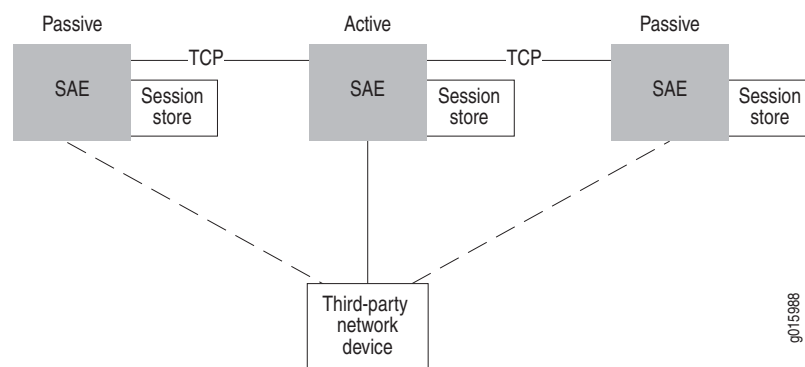
To log in subscribers to the SAE, you use assigned IP subscribers or event notification from an IP address manager.

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

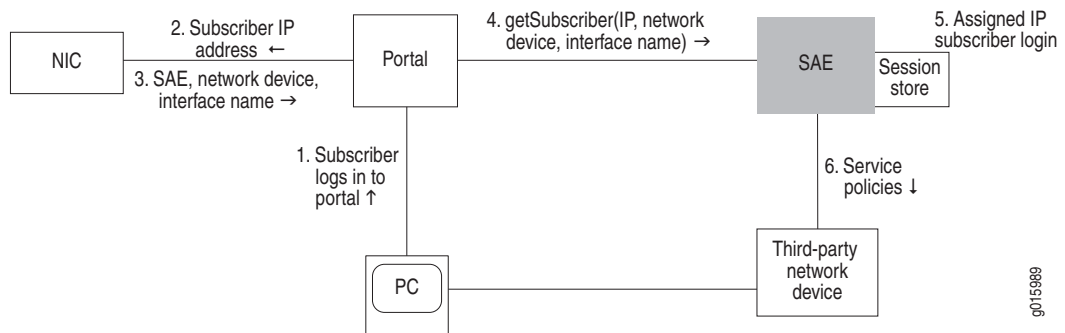
Assigned IP Subscribers

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

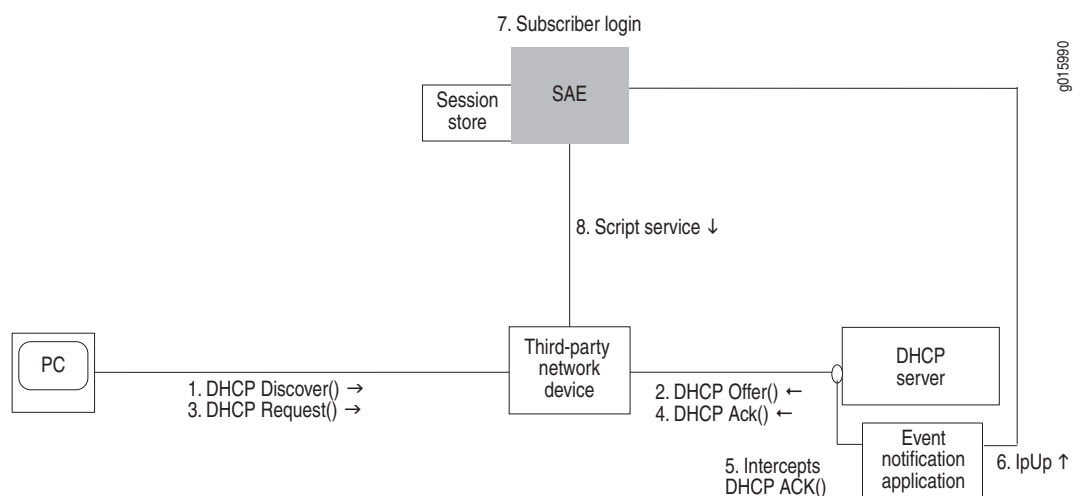
With the event notification method of logging in subscribers and creating subscriber sessions, the subscriber logs in to the network device and obtains an IP address through an address server, usually a DHCP server. The SAE receives notifications about the subscriber, such as the subscriber's IP address,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

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 *SRC-PE Network Guide: SAE, Juniper Networks Routers, NIC, and SRC-ACP, 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, a sample application that was created with the event notification API and that monitors DHCP or RADIUS messages for DHCP or RADIUS servers.

See the *SRC Sample Applications Guide*.

Setting Up Script Services

To set up script services:

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

See [SRC-PE 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 [SRC-PE 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-plugin [tracking-plugin...];
}
```

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-plugin [tracking-plugin...]
```

7. (Optional) Verify your configuration.

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

Setting Up SAE Communities

Tasks to configure SAE communities are:

- [Adding Virtual Router Objects on page 11.](#)
- [Configuring the SAE Community Manager on page 12.](#)
- [Specifying the Community Manager in the SAE Device Driver on page 14.](#)
- If there is a firewall in the network, configuring 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
<code>Ssp.properties</code>	System properties object (class: <code>java.util.Properties</code>)—The properties should be treated as read-only by the script.
<code>Ssp.errorLog</code>	Error logger—Use the <code>Ssp.errorLog.println (message)</code> to send error messages to the log.
<code>Ssp.infoLog</code>	Info logger—Use the <code>Ssp.infoLog.println (message)</code> to send informational messages to the log.
<code>Ssp.debugLog</code>	Debug logger—Use the <code>Ssp.debugLog.println (message)</code> to send debug messages to the log.

The 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.println("Setup connection to VR %(vrName)s" %
        vars(self))

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

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

```

Copying Initialization Scripts to the C-series Controller

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

```

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

```

Specifying Initialization Scripts on the SAE

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

```

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

```

To configure initialization scripts for third-party devices:

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

```

user@host# edit shared sae configuration driver scripts

```

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

```

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

```

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

```

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

```

4. (Optional) Verify your initialization script configuration.

```

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

```

Using SNMP to Retrieve Information from Network Devices

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

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 [SRC-PE Network Guide: SAE, Juniper Networks Routers, NIC, and SRC-ACP, Chapter 10, Configuring NIC with the SRC CLI](#) for information about configuring NIC scenarios.

Part 2

Integrating Directories

Chapter 2

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.

Topics in this chapter include:

- [LDAP Overview on page 23](#)
- [Supported Directories on page 25](#)
- [Directory Security on page 25](#)
- [Provisioning the Directory on page 26](#)
- [Naming Directory Entries on page 26](#)
- [SDX Directory Schema and Object Model on page 27](#)
- [Directory Schema for SRC Software on page 29](#)

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 [SRC-PE Getting Started Guide, Chapter 31, 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 [SRC-PE Getting Started Guide, Chapter 31, 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 6, Integrating the DirX Directory Server](#).
- eTrust Directory—See [Chapter 3, Integrating eTrust Directory](#).
- Oracle Internet Directory —See [Chapter 4, Integrating Oracle Internet Directory](#).
- Sun ONE Directory Server—See [Chapter 5, 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 25](#)
- [LDAPS Directory Connections on page 26](#)

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 *SRC-PE 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 8, 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 7, Configuring LDAPS for SRC Components](#).

Provisioning the Directory

You can provision the directory by:

- Using SRC applications such as SDX Admin 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.

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.

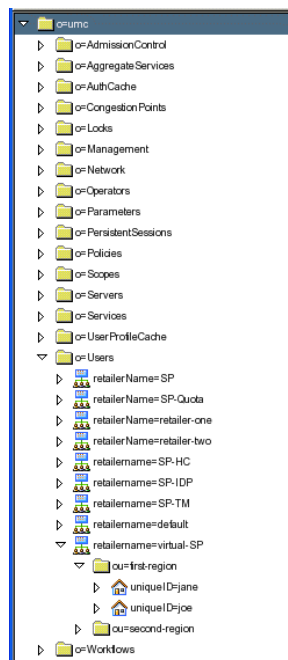
When you add an entry to the directory, do not use an asterisk (*) or other non-alphanumeric characters in the name. Do not specify object names that are the same but differ only in case use in the name. For example, do not use myrouter and MyRouter. Java applications and the enterprise service portals 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.

SDX Directory Schema and Object Model

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.

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 4](#) 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 4: 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.

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 4](#).

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 4](#):

```
o = umc
o = Users
retailname = 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, retailname = 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 5](#) lists some of the common DN attribute types.

Table 5: 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 for SRC Software

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

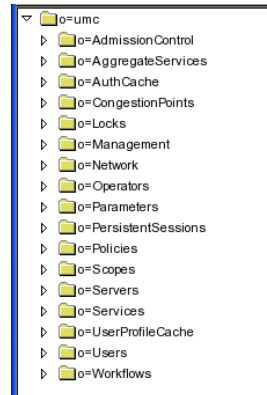
Object classes define the different types of entries that can exist in the directory; each entry in the directory belongs to one or more object classes. An object class contains specified attributes to define the characteristics of the entry. For information about attributes, see [Attributes on page 32](#).

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

- [Objects Representing Folders on page 29](#)
- [Subscriber Objects on page 30](#)
- [Service Objects on page 31](#)
- [Subscription Profile Objects on page 31](#)
- [Policy Objects on page 31](#)
- [Network Device Objects on page 31](#)
- [Configuration and System Management on page 32](#)

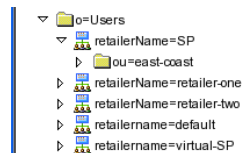
Objects Representing Folders

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 5](#) 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 5: First-Level Folders

The standard object classes organization and organizationalUnit divide trees into subtrees. In some cases, these object 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 6](#), the *ou = east-coast* folder is an organizationalUnit object.

Figure 6: 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.

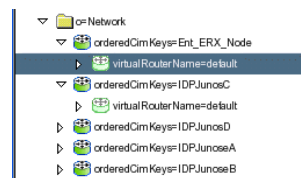
Network Device Objects

Physical network elements are modeled with the CIM Chassis object classes. The object class `d1m1Chassis` 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 7 shows a sample network folder that contains a number of network devices.

Figure 7: 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 `d1m1UnitaryComputerSystem` represents the hosts on which SAE and system management components, such as an SNMP server, are installed. For management components, the CIM object class `d1m1UnitaryComputerSystem` replaces the object class `umcHost`.

The `d1m1UnitaryComputerSystem` object class stores an IP address in the CIM attribute `d1m1IdentifyingDescriptions`. The location (for example, POP A) is specified in `d1m1OtherIdentifyingInfo`.

All configuration and management objects are stored under *o = Management*, *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, and user 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 6 lists the location of the LDIF files for the various directories.

Table 6: 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 3

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.

Topics in this chapter include:

- [Overview of Integration with eTrust Directory on page 35](#)
- [Integrating the eTrust Directory with the SRC Software on page 37](#)
- [Starting SDX eTrust Directory on page 38](#)
- [Stopping SDX eTrust Directory on page 39](#)
- [Displaying the Status of SDX eTrust Directory on page 39](#)
- [Backing Up and Restoring eTrust Directory on page 39](#)

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.dxc. 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 37](#)
2. [Configuring the SDX eTrust Directory Server Agent with the SRC Software on page 37](#)

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 [SRC-PE Getting Started Guide, Chapter 33, Installing the SRC Software on a Solaris Platform](#).

To integrate eTrust Directory:

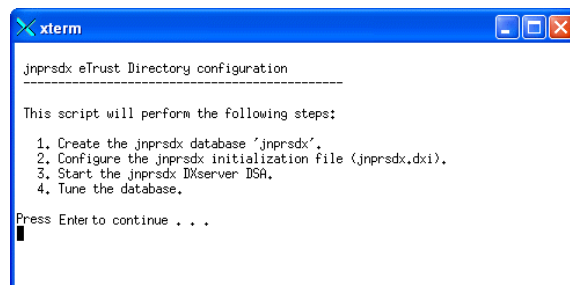
1. Navigate to the following directory:

`/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:



```

xterm
jnprsdxeTrust Directory configuration
-----
This script will perform the following steps:
1. Create the jnprsdxe database 'jnprsdxe'.
2. Configure the jnprsdxe initialization file (jnprsdxe.dxi).
3. Start the jnprsdxe DSA.
4. Tune the database.
Press Enter to continue . . .

```

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 35](#).

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 4

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.

Topics in this chapter include:

- [Overview of Oracle Internet Directory Integration on page 41](#)
- [About the Oracle Internet Directory Add-On Package on page 42](#)
- [Integrating the Oracle Internet Directory with the SRC Software on page 42](#)
- [Starting and Stopping Oracle Internet Directory on page 44](#)
- [Setting Up Local Configuration for SRC Components on page 45](#)
- [Backing Up and Restoring the Oracle Internet Directory on page 45](#)

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, use a supported RADIUS server. If you use Merit AAA Server, it 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 43](#)
2. [Running the Load Script for the Oracle Internet Directory Integration on page 44](#)

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 [SRC-PE Getting Started Guide, Chapter 33, 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 42](#).

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 5

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 47](#)
- [Integrating the Sun ONE Directory with the SRC Software on page 49](#)
- [Starting Sun ONE Directory Server on page 51](#)
- [Stopping Sun ONE Directory Server on page 52](#)
- [Restarting Sun ONE Directory Server on page 52](#)
- [Backing Up the Sun ONE Database on page 52](#)
- [Restoring the Sun ONE Database on page 53](#)

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 7](#) describes important setup script information provided in an *sdx.inf* file.

Table 7: Information Provided for the Sun ONE Setup Script

Configuration Property	Value
Installation path	<i>/opt/UMC/iDS</i>
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: <i>/opt/UMC/iDS/slaped-sdx</i>	sdx
Suffix for new LDAP directory instance	<i>o = umc</i>
Identifier for superadministrator	<i>cn = umcAdmin, o = umc</i>
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.

If you use Merit AAA Server (RADIUS), note that it 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 50](#)
3. [Configuring an Instance of Sun ONE Directory Server on page 50](#)

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 *SRC-PE Getting Started Guide, Chapter 33, 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.

```

Tera Term - 10.227.7.100 VT
File Edit Setup Control Window Help
adding new entry cn=ssp,ou=Components,o=Operators,o=UMC
adding new entry cn=osn,ou=Components,o=Operators,o=umc
modifying entry o=UMC
modifying entry o=Operators,o=UMC
modifying entry o=Users,o=UMC
modifying entry o=Policies,o=UMC
modifying entry o=Parameters,o=UMC
modifying entry o=Locks,o=UMC
modifying entry ou=System Management,ou=Configuration,o=Management,o=UMC
modifying entry o=Network,o=UMC
modifying entry o=Workflows,o=UMC
Restarting LDAP Server ...
Do you want to load sample data? [n]

```

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 [SRC-PE Getting Started Guide, Chapter 39, 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/slaped-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/slaped-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 [SRC-PE Getting Started Guide, Chapter 39, 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 6

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.

Topics in this chapter include:

- [Overview of DirX Directory Server Integration on page 55](#)
- [About the DirX Add-On Package on page 56](#)
- [Integrating the DirX Directory with the SRC Software on page 56](#)
- [Provisioning the Directory by Using DirXmetahub on page 59](#)
- [Uninstalling the DirX Directory Server on page 60](#)
- [Starting the DirX Directory Server on page 60](#)
- [Stopping the DirX Directory Server on page 61](#)
- [Backing Up the DirX Database on page 62](#)
- [Restoring the DirX Directory Database on page 62](#)

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 8](#) 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 8: Information Needed for DirX Generate Script

Configuration Property	Value
LDAP port to be used for directory instance	389
Suffix for new LDAP directory instance	<i>/o = umc</i>
Identifier for superadministrator	<i>/o = umc, cn = umcAdmin</i>
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 57](#)
2. [Installing the DirX Directory Server on page 58](#)
3. [Installing the UMCdirxa Add-On Package on page 58](#)
4. [Configuring the DirX Directory Server on page 58](#)

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 57](#), 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 9](#).

Table 9: Prompts for Installation of DirX-SV Server Package

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

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 [SRC-PE Getting Started Guide, Chapter 33, 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/dirxdapv3
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:
2. Change the directory to the *customize* directory:

```
login dirx
```

```
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 [SRC-PE Getting Started Guide, Chapter 39, 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 [SRC-PE Getting Started Guide, Chapter 39, 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 7

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.

Topics in this chapter include:

- [Overview of LDAPS Support on page 63](#)
- [LDAPS Authentication and Connection on page 63](#)
- [Configuring LDAPS Connections on page 64](#)

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 64](#)
2. [Establishing Trust for Directory Clients on page 65](#)
3. [Configuring the SAE to Find the Certificate Store on page 65](#)
4. [Enabling LDAPS Communication for SAE Components on page 66](#)
5. [Disabling LDAPS Communication for SAE Components on page 67](#)

Configuring the Directory Server to Support LDAPS Connections

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

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.

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 the `cacerts` file:

- View the file on the host on which you installed the JRE.

If your CA is not in the `cacerts` file:

- 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 10](#) shows how to access the security properties for the various components.

Table 10: How to Access Security Properties for SRC Components

SRC Component	How to Enable Security Properties
■ Configuration Manager	■ Edit the <code>/opt/UMC/sae/etc/default.properties</code> file.
■ User Data Manager	<ol style="list-style-type: none"> 1. In SDX Admin, select a configuration object (such as <code>I = POP_ID</code>) under <code>I = SAE</code>, <code>ou = staticConfiguration</code>, <code>ou = Configuration</code>, <code>o = Management</code>, <code>o = umc</code>. 2. Select the Main tab. <p>The security properties appear in the list of properties on the Main tab.</p>
■ Equipment Data Manager	
■ Service Data Manager	
■ LDAP Authentication Plug-in	
■ License Manager	
■ Enterprise Service Portal User Data Manager	<ol style="list-style-type: none"> 1. In SDX Admin, select <code>I = EASP</code>, <code>ou = staticConfiguration</code>, <code>ou = Configuration</code>, <code>o = Management</code>, <code>o = umc</code>. 2. Select the Main tab. <p>The security properties appear in the list of properties on the Main tab.</p>
■ Enterprise Service Portal Service Data Manager	

2. Remove the comment character (#) that appears before the component's security protocol property. See [Table 11](#).

Table 11: Security Protocol Properties for SAE Components

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

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 12](#)) to the value supported for the LDAPS connection.

Table 12: 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 10](#).
2. Add a comment character before the component's security protocol property. See [Table 11](#).
3. Set the server port property (as listed in [Table 12](#)) to the value supported for the LDAP connection.

For LDAP connections, the default port number is 389.

4. Save the configuration.

Chapter 8

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*.

Topics in this chapter include:

- [Directory Configuration on page 69](#)
- [Directories on page 70](#)
- [User Class on page 70](#)
- [Permissions on page 70](#)
- [Access Controls on page 71](#)
- [Directory-Specific Access Control Implementation on page 81](#)

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 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 $o = \text{Operators}$, $o = \text{umc}$
- Read access to the remaining directory tree, including the operational attributes `creationTimeStamp` and `modifyTimeStamp`
- No read and compare rights for any `userPassword` values

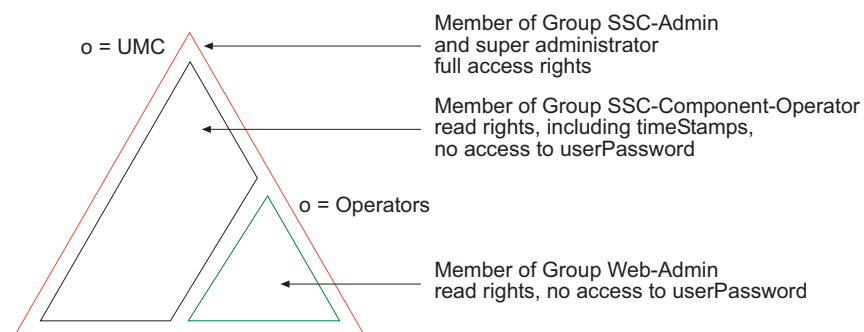
Clients binding with the Apache DN or a member of the WebAdmin group do have read and search permissions in the subtree $o = \text{Operators}$, $o = \text{umc}$:

- 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 8: Access Rights for the UMC Tree

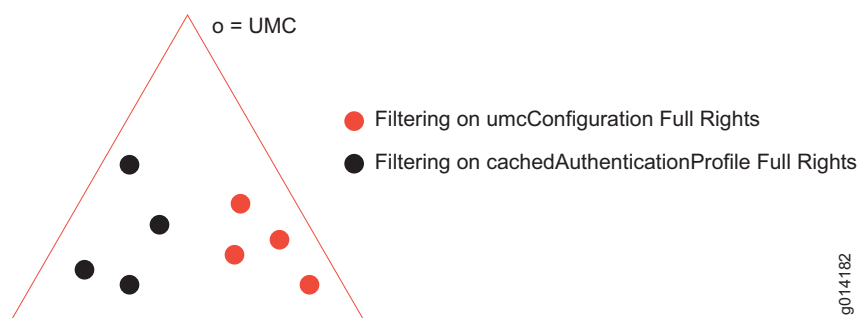


Access Controls Against Objects from Type `cachedAuthenticationProfile` and `umcConfiguration`

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`).

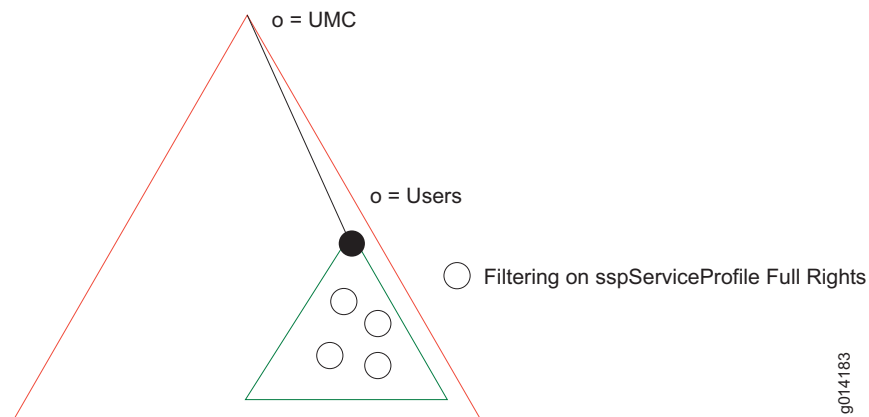
Figure 9: Access Rights Against `cachedAuthenticationProfile` and `umcConfiguration` Objects



Access Controls Against `sspServiceProfile`

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

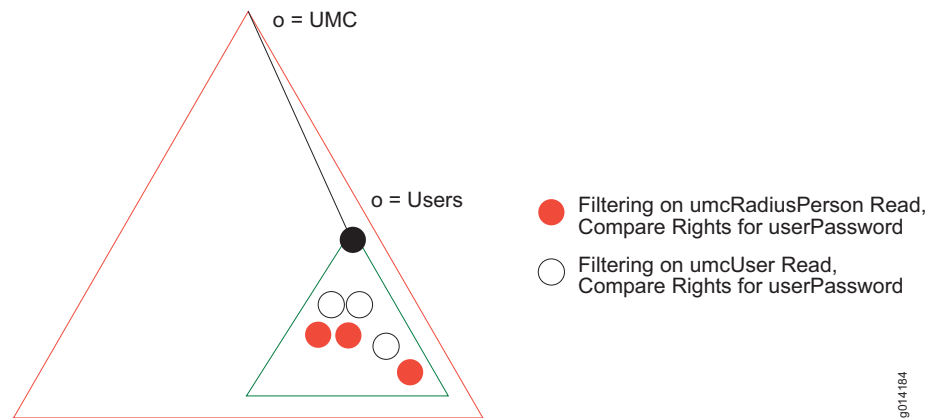
Figure 10: Access Controls Against `sspServiceProfiles` in the User Subtree



Access Controls Against umcRadius Person and umcUser

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

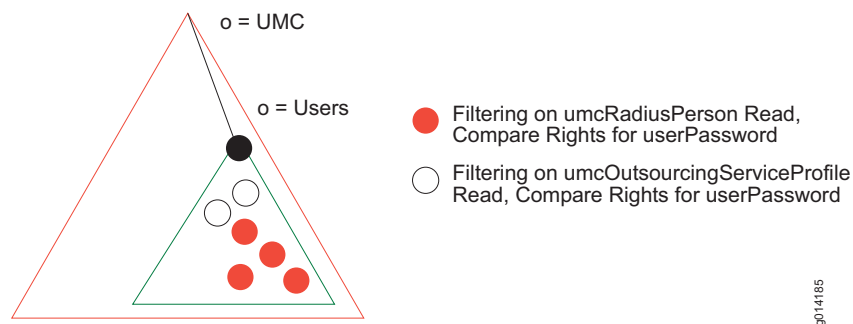
Figure 11: Access Rights Against umcRadiusPerson and umcUser



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*.

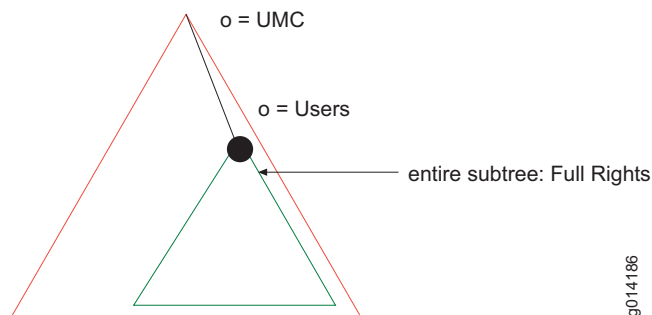
Figure 12: Access Rights Against umcRadiusPerson and umcOutsourcingServiceProfile Objects



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.

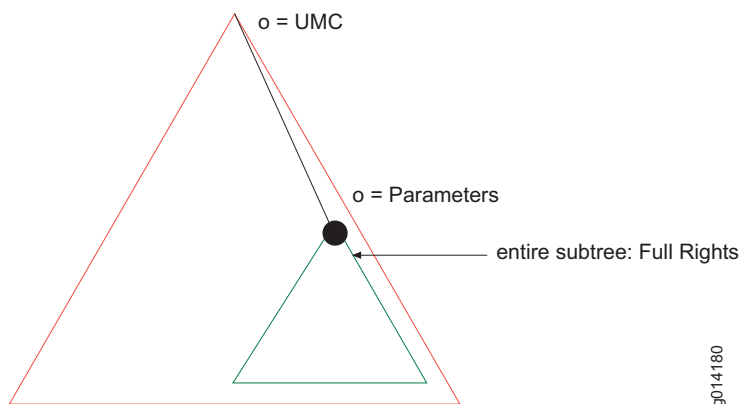
Figure 13: Policy Rights Against All Objects in the *o=Policies,o=umc* Tree



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.

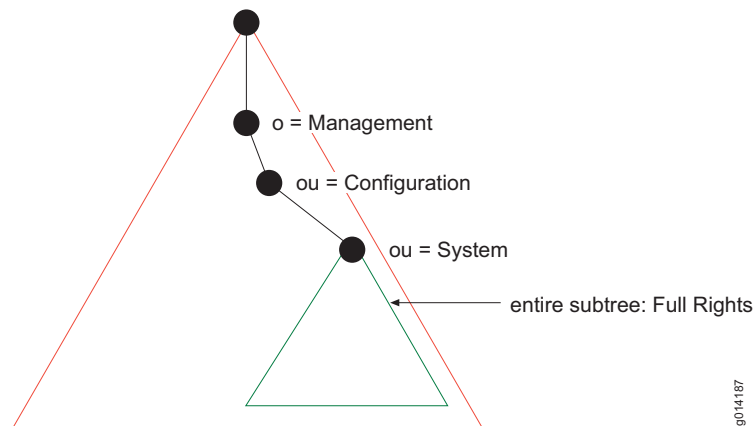
Figure 14: Access Rights Against All Objects in the Tree *o=Parameters, o=umc*



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*.

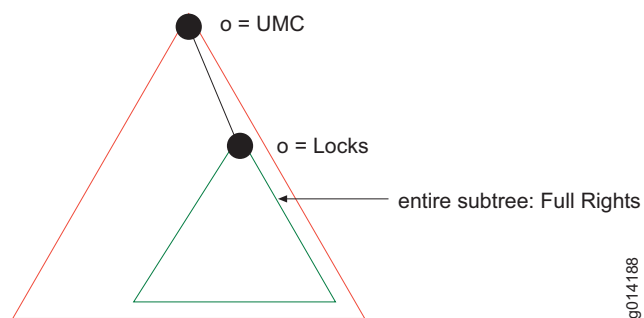
Figure 15: Access Rights for System Management



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.

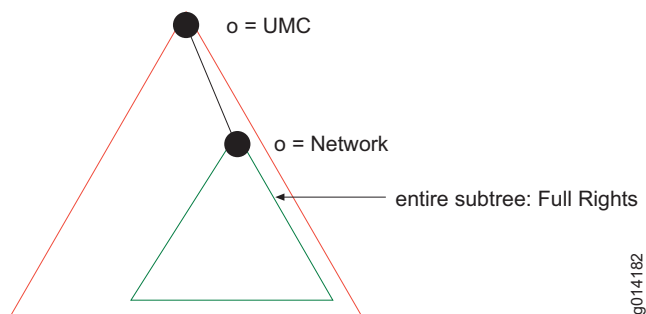
Figure 16: Access Rights Against the Entire *o=Locks,o=umc* Subtree



Access Controls Against the Network Subtree

The network operator is allowed to administer only objects within the subtree $o = \text{Network}$, $o = \text{umc}$ and bind against the directory using the credentials of $cn = \text{network-operator}$, $o = \text{operators}$, $o = \text{umc}$.

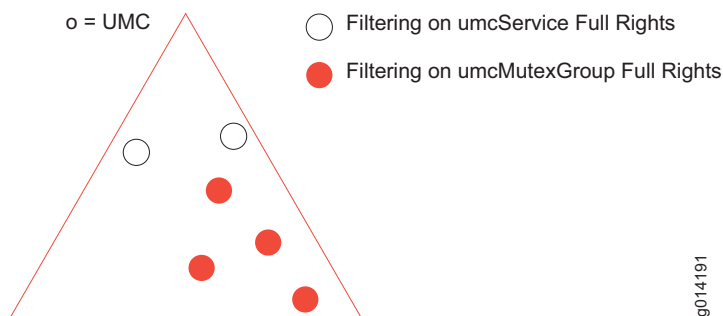
Figure 17: Access Rights Against the Entire $o = \text{Network}, o = \text{umc}$ Subtree



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 $o = \text{Services}$, $o = \text{umc}$ and $o = \text{Scopes}$, $o = \text{umc}$. The service-operator binds with the DN $cn = \text{service-operator}$, $o = \text{operators}$, $o = \text{umc}$ against the directory.

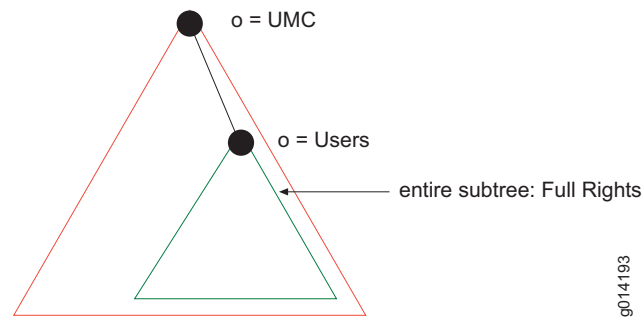
Figure 18: Access Rights Against `umcService` and `umcMutexGroup` Objects



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 $cn = subscriber-operator$, $o = operators$, $o = umc$.

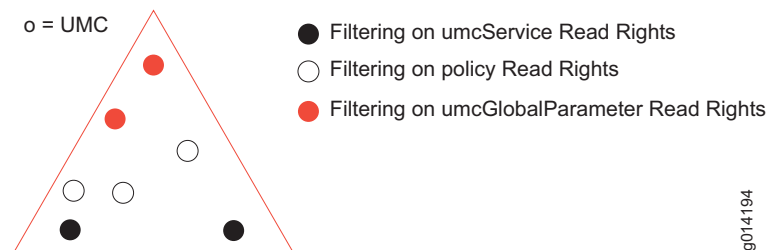
Figure 19: Access Rights Against the Entire $o=users$, $o=umc$ Subtree



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.

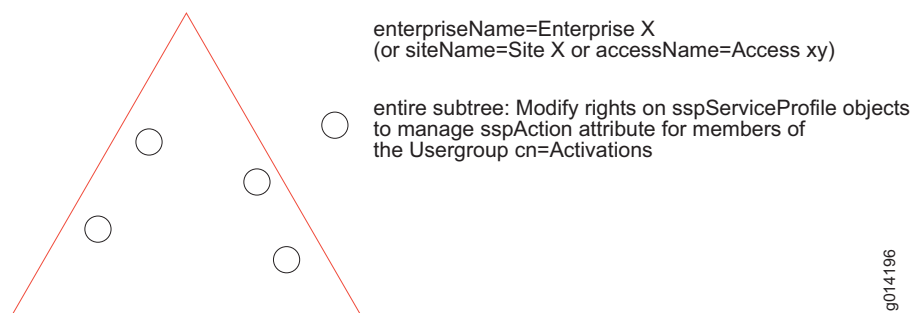
Figure 20: Access Rights Against `umcService`, `Policy`, and `umcGlobalParameter` Objects



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 21](#) shows these modify rights.

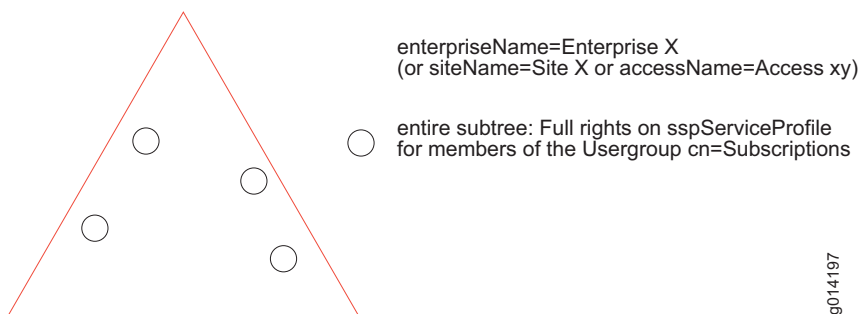
Figure 21: Modify Rights for Activation Managers



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 22](#).

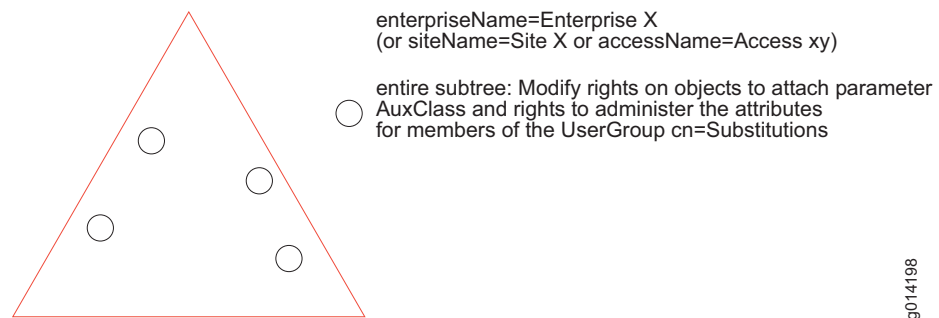
Figure 22: Access Rights for Subscription Managers



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 23: Access Rights for Substitution Managers

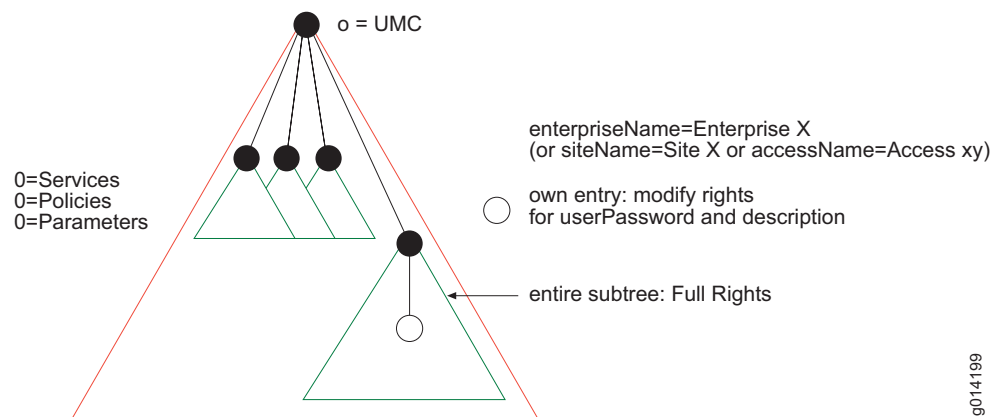


Common Access Rights for All Managers

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 24: 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 25 shows a TCL script with an explanation of the various parts.

Figure 25: Creation of an Access Control Subentry Example in DirX

```
create /o=UMC/CN=SSP-AccessControl-Subentry \ 1
{OCL=SUBE;ACS} \
{SS={SF={OR={ITEM=authProfile}; \ 2
{ITEM=umcConf} } }}\
3 PACI={ID=SSP: Full rights on Cached Profiles and Configuration;
PR=254, 4
5 AL={BL={L=SIMPLE}},
UF={UC={N={DN={/o=UMC/o=Operators/ou=Components/cn=ssp}}},
7 UP={PI={E=TRUE}, GAD=$AEM}; 8 6
{PI={AUATV=TRUE}, GAD=$AAM} } 9 10
```

1. DN of subentry
2. Target (entire area)
3. (one or more) Identifier(s) of Prescriptive ACI
4. Precedence [0-255]
5. Authentication-level simple-bind
6. User-class: SSP component
7. First protected items (all entries)
8. Grant and denials for all entries: Full Rights
9. Second protected items (all user-attributes)
10. Grant and denials for all user-attributes: Modify Rights

g014955

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 26 shows the LDIF file for implementing the same kind of access level as previously depicted with a Sun ONE directory.

Figure 26: 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 9

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. 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.

Topics in this chapter include:

- [System Requirements for the Steel-Belted Radius Server on page 87](#)
- [Installing the Steel-Belted Radius/SPE Software on page 87](#)

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

You can find information about the system requirements for the Steel-Belted Radius server in the release notes at the following Web site:

http://www.juniper.net/techpubs/software/aaa_802/sbr.html

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.

You must have JNPRsbrsp.pkg and UMCsbrspa.pkg to install and integrate the Steel-Belted Radius/SPE software. Further information about the installation and configuration procedures is located at the following Web site:

http://www.juniper.net/techpubs/software/aaa_802/sbr.html

Chapter 10

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.

Topics in this chapter include:

- [System Requirements for the RAD-Series RADIUS Server on page 90](#)
- [Installing the RAD-Series RADIUS Server on page 90](#)
- [LDAP Features for the RAD-Series RADIUS Server on page 91](#)
- [Configuring UDP Ports for the RAD-Series RADIUS Server on page 92](#)
- [Starting and Stopping RAD-Series Server Manager on page 93](#)
- [Extending Dictionary Files with JUNOSe Parameters for the RAD-Series RADIUS Server on page 95](#)
- [Configuring LDAP Authentication for the RAD-Series RADIUS Server on page 95](#)
- [Example: RAD-Series RADIUS Server Accounting Log File Format on page 100](#)
- [Configuring the RAD-Series RADIUS Server and RADIUS Clients on page 101](#)
- [Testing the RAD-Series RADIUS Server on page 102](#)

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 95](#).

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 JUNOS 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 JUNOS 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 93](#).

Starting and Stopping RAD-Series Server Manager

To open RAD-Series Server Manager:

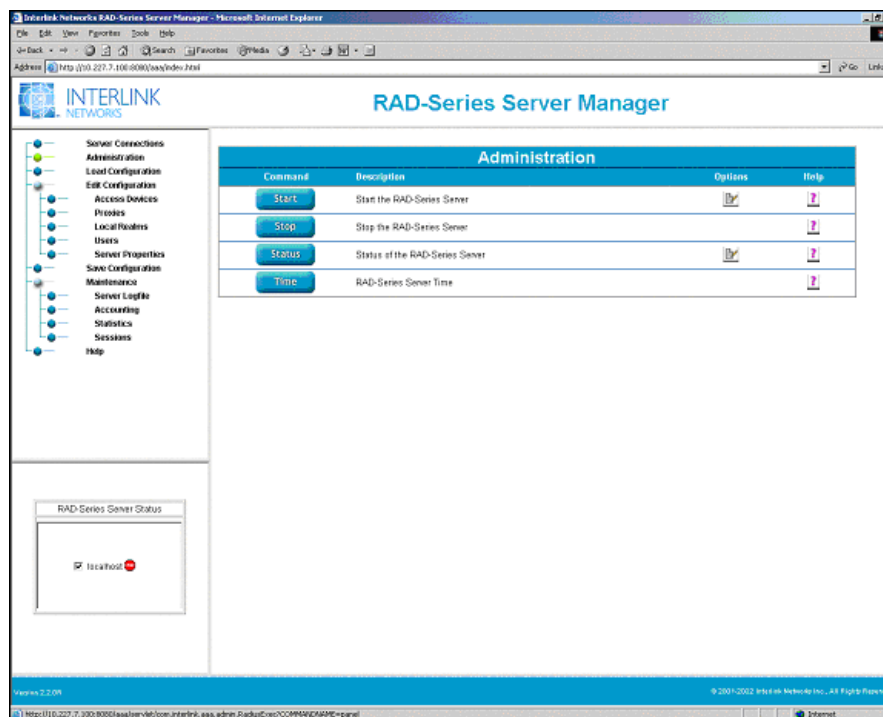
1. Start Tomcat by entering:
/opt/aaa/tomcat/bin/startup.sh
2. Enter the following URL into your Web browser:
http://<ip-address of server>:8080/aaa/index.html
3. When prompted for the Server Manager username, enter:
admin
4. When prompted for the Server Manager password, enter:
radius



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.



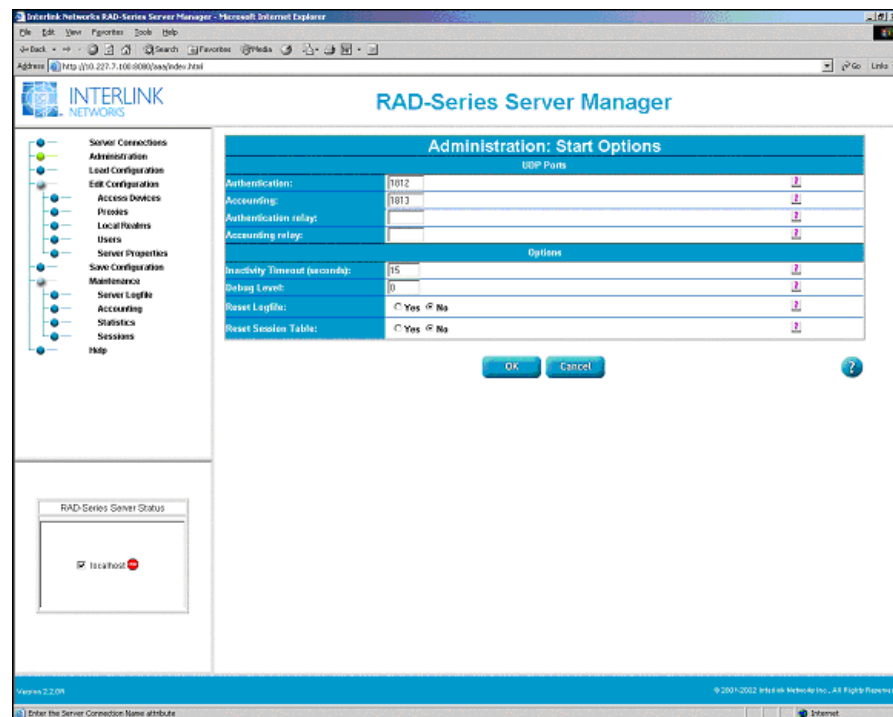
To start or stop RAD-Series Server Manager, click the Start or Stop button. When RAD-Series Server Manager is running, the bullet in the navigation pane turns green. When RAD-Series Server Manager is not running, it is blue.

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 92](#):

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.

Tasks to configure LDAP authentication for the RAD-Series RADIUS Server are:

- [Configuring the RAD-Series Server Manager on page 96](#)
- [Configuring Realm Administration on page 98](#)
- [Configuring LDAP Settings on page 99](#)
- [Configuring RADIUS Profiles with the LDAP Directory on page 99](#)

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

```

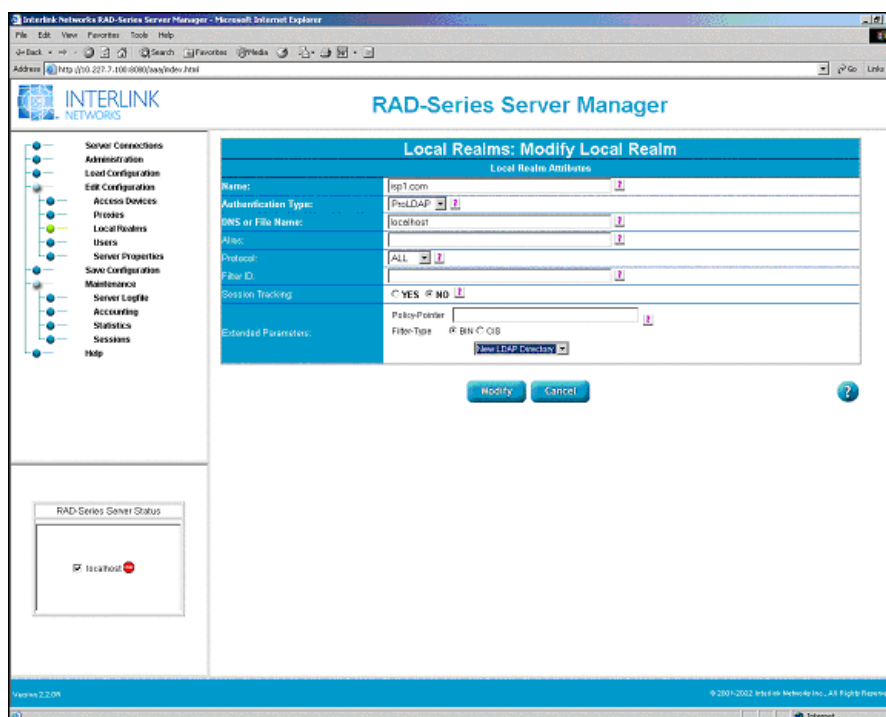
Configuring Realm Administration

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 JUNOS-specific attributes must be returned to the JUNOS 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 JUNOS-specific attribute for the virtual router to be used on the JUNOS 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 JUNOS 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:

```
radpwst -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
 OID – Oracle Internet Directory
 RAD – RAD-Series AAA RADIUS Server
 SPE – Steel-Belted Radius/SPE
 Sun – Sun ONE Directory Server

A

aaaCheck, RAD 95
 aaaDeny, RAD 88, 95
 aaaReply, RAD 88, 95
 access control information
 DirX directory server 54
 eTrust Directory 34
 Oracle Internet Directory 40, 43
 Sun ONE Directory Server 47
 access control scheme
 activating access
 operator 76
 subscription operator 76
 binding 69
 cachedAuthentication profile 70
 common access rights 77
 directories 68
 directory entries 67
 directory-specific 78
 DirX directory server 78
 lock subtree 73
 mutex group objects 74
 network subtree 74
 parameter subtree 72
 permissions 68
 policy subtree 72
 RADIUS profile 71
 service, policy, and global parameter objects 75
 sspServiceProfile 70
 substitution access 77
 Sun ONE Directory Server 79
 system management 73
 UmcConfiguration 70
 umcRadius Person 71
 umcUser 71
 user class 68
 user subtree 75
 accounting log file, RAD 96

aci attribute (Sun) 47
 add-on packages
 DirX directory server 54
 eTrust Directory 33
 Oracle Internet Directory 40
 Sun ONE Directory Server 46–47
 applications
 SRC on CD xi
 assigned IP subscribers
 third-party devices 6
 IP address pools 6
 attribute value pairs, RAD 88
 attributes
 AAA, RAD 95
 JUNOS-specific, RAD 91, 95, 96
 LDAP 30
 audience for documentation ix
 authentication/authorization transfer vectors (AATVs),
 RAD 87
 authfile file, RAD 92

B

backing up
 DirX directory 60
 Sun ONE directory 50
 binary directory, RAD 86
 binding
 access control 69

C

cacerts files, LDAPS 63
 cachedAuthentication and access controls 70
 certificate files, LDAPS 63
 Class, RAD 96
 commands
 keytool 63
 Common Information Model (CIM) 25
 community manager
 configuring, third-party devices
 SRC CLI 12
 configuration directory, RAD 86
 content rules 31
 conventions defined
 icons x
 text x

customer support xiv

D

data directory, RAD 86

dictionary files with JUNOS parameters, RAD 91

directory

access control scheme 67–80

access rights 23

provisioning 57

redundancy 22

schema 25–31

supported directory servers 23

updates to 22

See also LDAP

directory server

See DirX directory server; eTrust Directory; Oracle

Internet Directory; Sun ONE Directory Server

DirX directory server

access control scheme 78

add-on package 54

backing up directory 60

directory user 55

dirx user environment 58, 59

installing 56

integrating with SRC software 54–57

loading sample data 57

obtaining 55

restoring directory database 60

standards 54

starting 58, 59

stopping 59

superuser environment 59

uninstalling 58

DirXmetahub 54, 57

DIT (Directory Information Tree)

content rules 31

structure rules 31

d1m1Chassis object class 29

documentation set, SRC. *See* SRC documentation set

E

eTrust Directory

add-on package 33

installing 35

integrating with SRC software 35

Juniper SDX eTrust Directory

displaying status 37

starting 36

stopping 37

load script 33, 36

loading sample data 36

setup.sh script 34, 35

event notification, third-party devices

configuring properties

SRC CLI 14

description 7

G

generate.sh script (DirX) 57

global parameter objects and access

control scheme 75

H

hard-disk space requirements, RAD 86

I

icons defined, notice x

Idle-Timeout 96

initialize.cp flat file (DirX) 54

installation

software CD (SPE) 83

installing software

RAD 86

SPE 83

integrating data into directory 24

J

Juniper Networks dictionary files, RAD 91

Juniper SDX eTrust Directory. *See* eTrust Directory

JUNOS RADIUS client, RAD 96

K

keytool command 63

L

LDAP

attributes 30

integration overview 21–23

See also directory; LDAPS

LDAP authentication

AATV, RAD 87

LDAP configuration, RAD 95

LDAP features, RAD 87

LDAPS

authentication and connection sequence 61

cacerts files 63

certificate files 63

configuring

directory server 62

overview 62

SAE components 64, 65

directory connections 24

disabling connection for SAE components 65

enabling connection for SAE components 64

overview 61

LDIF (LDAP Data Interchange Format) files 24, 32

lock subtree and access control scheme	73
log.txt log file (DirX)	57
login process	
assigned IP subscribers, third-party devices	6
event notification method, third-party devices	7

M

manuals, SRC	
comments	xiii
Merit AAA server	
Oracle Internet Directory integration, with	39
meta agent	57
metacontroller	57
metadirectory store	57
mutex group	
access control scheme	74

N

naming syntax	24
network and access control scheme	74
notice icons defined	x

O

object classes	27
operating system requirements, RAD	86
operators	
activating access	76
Oracle Internet Directory	
add-on package	40
directory settings	42
installing	41
integrating with SRC software	40–42
load script	40, 42
loading sample data	42
Merit AAA server, and	39
password configuration	40

P

parameters	
subtree and access control scheme	72
PCIM (Policy Core Information Model)	29
permissions in access control lists	68
policy objects	
access control	75
policy subtree and access control scheme	72
ProLDAP AATV, RAD	88, 92

R

RADIUS	
profiles and access control scheme	71
RADIUS attributes	
dictionary files, RAD	91
RADIUS authentication	
RAD	91

RADIUS client	
configuration, RAD	98
UDP ports, RAD	88
vendor, RAD	97
RADIUS profiles, RAD	95
RADIUS shared secret, RAD	97, 98
RAD-Series RADIUS Server	
configuring	97
installation	86
starting	89
stopping	89
syntax description	98
testing	98
testing installation	98
RAD-Series Server Manager	89
RAM requirements, RAD	86
realm	
administration, RAD	94
realm name	
authentication, RAD	92
description, RAD	92
release notes	xiii
restoring	
DirX directory database	60
Sun ONE directory database	51
router object	
adding for third-party devices	
SRC CLI	10
S	
SAE (service activation engine)	
security properties	64
SAE communities	
configuring, third-party devices	
SRC CLI	12
description, third-party devices	4
sample data	32
loading	
DirX directory server	57
eTrust Directory	36
Oracle Internet Directory	42
Sun ONE Directory Server	49
schema, loading	
DirX directory server	54
Oracle Internet Directory	40
Sun ONE Directory Server	47
script services	
for third-party devices	5
scripts	
generate (DirX)	54
load (eTrust Directory)	33
load (OID)	40, 42
load (Sun)	49
setup.sh (eTrust Directory)	34

Server Manager		
password, RAD	87	
user, RAD	87	
service objects, access control	75	
session store		
in third-party networks	4	
Session-Timeout, RAD	96	
shared secret, RAD	87	
silent installation file (Sun)	46	
SNMP		
retrieving information from network devices	18	
software		
CD installation (SPE)	83	
SRC documentation set		
comments	xiii	
obtaining	xiii	
SRC documentation CD	xi	
SRC software distribution	xiii	
SSL (secure sockets layer). <i>See</i> LDAPS		
sspServiceProfile and access control scheme	70	
structure rules	31	
subscriptions		
activating access for operators	76	
substitutions		
access	77	
Sun ONE Directory Server		
access control	79	
add-on package	46–47	
backing up directory	50	
installing	48	
integrating with SRC software	47–49	
load script	49	
loading sample data	49	
obtaining	47	
product description	45	
restarting	50	
restoring directory	51	
silent installation feature	46	
starting	49	
stopping	50	
superuser environment (DirX)	59	
support, requesting	xiv	
syntax for directory entries	24	
system management		
access control scheme	73	
system requirements		
RAD	86	
T		
technical support, requesting	xiv	
test user password, RAD	87	
text conventions defined	x	
third-party devices		
creating sessions	5	
initialization scripts		
SRC CLI	15	
integrating into SRC network		
SRC CLI	3–18	
logging in subscribers		
assigned IP method	6	
event notification method	7	
overview	5	
provisioning with script services	5	
router objects, adding		
SRC CLI	10	
SAE communities	4	
VR objects, adding		
SRC CLI	11	
Tomcat installation, RAD	86	
transaction-based RADIUS protocol, RAD	88	
U		
UDP ports		
RAD	88, 90, 98	
uid-uniqueness plug-in (Sun)	46	
UmcConfiguration and access controls	70	
umcRadius Person and access control scheme	71	
umcUser and access control scheme	71	
Unisphere VSA, updating, RAD	91	
user class in access control lists	68	
user subtree and access control scheme	75	
V		
variables.tcl file (DirX)	54	
vendor files		
RAD	91	
vendor files, RAD	91	
virtual routers		
adding for third-party devices		
SRC CLI	11	
VSAs (vendor-specific attributes)		
RAD	91	