



**Juniper Networks
Steel-Belted Radius**

**SIM Server
Administration Guide**

*Release 5.4
December 2006*

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

Part Number: 54A-061212-KG Revision A

Copyright © 2005–2007 Juniper Networks, Inc. All rights reserved. Printed in USA.

Steel-Belted Radius, Juniper Networks, and the Juniper Networks logo are registered trademarks of Juniper Networks, Inc. in the United States and other countries. Ulticom, Signalware, Programmable Network, Ultimate Call Control, and Nexworx are registered trademarks of Ulticom, Inc. Kineto and the Kineto Logo are registered trademarks of Kineto Wireless, Inc. Software Advancing Communications and SignalCare are trademarks and service marks of Ulticom, Inc. Raima, Raima Database Manager and Raima Object Manager are trademarks of Birdstep Technology. Sun, Sun Microsystems, the Sun logo, Java, Solaris, and all trademarks and logos that contain Sun, Solaris, or Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners. All specifications are subject to change without notice.

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.

Portions of this software copyright 1989, 1991, 1992 by Carnegie Mellon University Derivative Work - 1996, 1998-2000 Copyright 1996, 1998-2000 The Regents of the University of California All Rights Reserved Permission to use, copy, modify and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU and The Regents of the University of California not be used in advertising or publicity pertaining to distribution of the software without specific written permission.

CMU AND THE REGENTS OF THE UNIVERSITY OF CALIFORNIA DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL CMU OR THE REGENTS OF THE UNIVERSITY OF CALIFORNIA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM THE 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.

Portions of this software copyright © 2001-2002, Networks Associates Technology, Inc All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Networks Associates Technology, Inc nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions of this software are copyright © 2001-2002, Cambridge Broadband Ltd. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

The name of Cambridge Broadband Ltd. may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions of this software copyright © 1995-2002 Jean-loup Gailly and Mark Adler This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

This notice may not be removed or altered from any source distribution.

HTTPClient package Copyright © 1996-2001 Ronald Tschalär (ronald@innovation.ch).

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. For a copy of the GNU Lesser General Public License, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

StrutLayout Java AWT layout manager Copyright © 1998 Matthew Phillips (mpp@ozemail.com.au).

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details. For a copy of the GNU Lesser General Public License, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

M061212

Table of Contents

About This Guide	ix
Audience	ix
What's In This Manual	ix
Conventions	x
Related Documentation	xi
Ulticom Documentation.....	xi
IETF Documentation.....	xii
Chapter 1 Overview	1
Steel-Belted Radius/SIM Server Components	1
Hardware Components.....	1
Software Components	2
Web-Based Authentication Using SMS.....	4
SMS Authorization and Account Provisioning	6
SMS Authentication	8
CDR Accounting and Billing.....	9
Fraud Prevention.....	10
SIMAuth Support	10
SIM Card-Based Authentication.....	12
EAP-SIM/EAP-AKA Authorization/Service Delivery	14
EAP-SIM/EAP-AKA Identities	14
EAP-SIM/EAP-AKA Fast Reauthentication	15
Chapter 2 System Setup	17
System Requirements	17
Installing and Running the Configuration Script	17
Installing Steel-Belted Radius/SIM Server	18
Running the Configuration Script.....	19
Starting and Stopping Steel-Belted Radius/SIM Server	22
Correlating Related Log Entries (Log Thread)	23
Installing Signalware 9.0	23
Signalware Packages	33
Removing Signalware	33
Starting and Stopping Signalware.....	33
Configuring Signalware to Start Automatically on Reboot	33
Starting and Stopping Signalware	34
Upgrading Solaris, Signalware, and SIM Server	35
Save the Existing Steel-Belted Radius/SIM Server Configuration.....	35
Save the Signalware 8 Configuration	36
Uninstall and Remove the Existing Version of SIM Server	36
Uninstall and Remove Signalware 8	37
Upgrade the Operating System from Solaris 8 to Solaris 9.....	38
Upgrading from Signalware 8 to Signalware 9	38

	Apply the Saved Signalware Configuration to Signalware 9	38
	Install Steel-Belted Radius/SIM Server 5.4	39
	Apply the Saved SIM Server Configuration to SIM Server 5.4.....	39
Chapter 3	Configuring SS7/IP Networks	41
	Communication Pathways and Corresponding Files	42
	Signalware MML Commands.....	42
	Configuration Activities.....	43
	Defining Links, Link Sets, and Route Sets.....	43
	Example MML Commands.....	44
	Configuring the authGateway Application for HLR Communication	46
	Configuring the authGateway Routing Location Information.....	46
	Configuring the authGateway.conf File	47
	Configuring the authGateway Startup Information.....	50
	Configuring the ulcmmg.conf File.....	52
	Configuring the SMSGateway Application for MSC Communication	52
	Configuring the SMSGateway Routing Location Information.....	53
	Configuring the SMSGateway.conf File	53
	Configuring the SMSGateway Startup Information	55
	Configuring the smsulcmmg.conf File	58
	Loading the MML Configuration Settings.....	58
Chapter 4	Configuring the Steel-Belted Radius Files	59
	Configuring the gsmmap.gen File	59
	[Bootstrap] Section	59
	[Settings] Section.....	60
	[Realms] Section.....	60
	Configuring Each Realm Section.....	61
	Network Equipment and Data Needed for Processing Access-Requests ..	63
	Target Module Section	64
	Configuring sqlaccessor.gen and sqlaccessorjdbc.gen	67
	[Bootstrap] Section	68
	[Settings] Section.....	68
	SQL Server Connection Methods.....	70
	Connecting to a Single SQL Server.....	70
	Connecting to Multiple Servers	71
	SQL Database Data Retrieval Methods	72
	SQL = SELECT Method for Data Retrieval from SQL Databases.....	72
	Stored Procedure Method for Data Retrieval from SQL Databases.....	74
	Configuring the LDAP Data Accessor (ldapaccessor.gen).....	76
	Identifying Key Fields for Oracle, JDBC, and LDAP Databases	82
	Configuring gsmmap.gen for Key Field Identification.....	82
	Configuring sqlaccessor.gen or sqlaccessorjdbc.gen for Key Field Identification.....	82
	Configuring ldapaccessor.gen for Key Field Identification.....	83
Chapter 5	Special Attribute Handling	85
	Adding Location Information to Access-Requests.....	85
	Overview	85
	Location-Specific Configuration Files	86
	locspectrnl File.....	87
	proxy.ini File	89
	realm.pro File	90

	Example Configuration for Adding NAS Location to Access-Request.....	90
	Assigning IP Addresses Based on Access Point Name (APN)	92
	Overview	92
	Tasks for Assigning IP Address Based on Access Point	93
	Configuring simauth.aut for IP Address Assignment	93
	Creating Address Pools	95
	Adding Attributes to an Access-Accept	96
	Overview	96
	Data Flow	97
	Configuration Tasks	98
	Configuring Files for Adding Attributes to Access-Accept	98
	Example Configuration for Adding Attributes to Access-Accept	100
	Activating Authentication	103
Chapter 6	Configuring the Call Detail Record Accounting Module	105
	Overview of CDR Process	105
	Types of Call Detail Records.....	105
	Configuring Accounting Options with cdracct.acc	106
	[Bootstrap] Section of cdracct.acc.....	106
	[Settings] Section of cdracct.acc	107
	Displaying CDR Information	110
	CDR Files.....	110
	Using cdrdump to Display CDR File Contents	111
	cdrdump Output.....	113
	Displaying ASN1 CDR Files in Raw Format Using dumpasn1	113
	CDR Fields	113
	CDR Field Formats for Binary and ASN.1 CDR Files	121
	Field Formats for Binary Version 1 and Binary Version 2 CDR Files	121
	Field Formats for ASN.1 CDR Files	122
Chapter 7	Configuring EAP-SIM/EAP-AKA Authentication	123
	Configuring the simauth.aut File	123
	simauth.aut [Bootstrap] Section.....	123
	simauth.aut [Settings] Section	124
	simauth.aut [ProfileMap] Section.....	127
Chapter 8	Configuring Web-Based Authentication with SMS	129
	Configuring SMS Options	129
	Configuring Account Provisioning Options	130
	Configuring SMS Authentication Options	133
	Configuring SMS Message Options.....	135
	Setting Up Message Templates	137
	SMS Interface Requirements	139
	Overview	139
	Access-Request	140
	Access-Request	141
	Interaction Examples.....	143
Appendix A	SS7 Sample Configuration Files	145
	Basic Provisioning MML File with One Point Code/ Two SS7 Links.....	145
	Routing Based on PC/SSN	146
	Redundant SS7 Links Backing Up Each Other (Two Point Codes).....	146

	authGateway Commands and Files	147
	AS4StartMapGw.mml — One Point Code/ Two SS7 Links	147
	AS4StartMapGw.mml — Routing Based on PC/SSN	147
	authGateway.conf (Gateway routing configuration file).....	148
	ulcmmg.conf	149
	smsGateway Commands and Files.....	149
	AS4StartSmsGw.mml — One Point Code/ Two SS7 Links	149
	smsGateway.conf	150
	smsulcmmg.conf	151
Appendix B	Reloading the Configuration with HUP Signals	153
Appendix C	SNMP Traps	159
Appendix D	Internal LDAP Directory	161
Appendix E	Kineto INC/SIM Server S1 Interface	163
	Attribute Handling Overview.....	163
	Access-Request Conversion.....	164
	Access-Accept Conversion.....	166
	Access-Reject Conversion.....	168
	Configuring SIM Server for Kineto Attribute Handling	168
	Configuring the kinetoUMAAAttrHandler.ctrl File.....	168
	Configuring the controlpoints.ini File.....	169
	Configuring Steel-Belted Radius to Recognize the Kineto Attributes.....	170
	Developing Applications for the S1 Interface	172
	Example Files.....	173
Appendix F	Glossary	175
	Index	181

About This Guide

Steel-Belted Radius/SIM Server enables GSM (Global System for Mobile Communications) and UMTS (Universal Mobile Telecommunications System) mobile service providers to offer wireless network access to subscribers through hotspot concession operators while leveraging existing customer care, roaming, and billing infrastructures.

The *Steel-Belted Radius/SIM Server Administration Guide* describes how to install, configure, and administer Steel-Belted Radius/SIM Server.

Audience

This manual is intended for network administrators who are responsible for implementing and maintaining authentication, authorization, and accounting services for a provider. This manual assumes that you are familiar with RADIUS, SS7 (Signaling System 7), SMS (Short Message Service), HLR (Home Location Register), GSM, UMTS, CDRs (Call Detail Records), and general networking concepts. This manual assumes that you are also familiar with the specific environment in which you are installing Steel-Belted Radius and the Ulticom Signalware platform.

What's In This Manual

This manual contains the following chapters and appendixes:

- Chapter 1, “Overview,” presents an overview of the components and operation of Steel-Belted Radius/SIM Server.
- Chapter 2, “System Setup,” describes how to set up Steel-Belted Radius/SIM Server and Ulticom Signalware on a “clean” system and how to upgrade from earlier versions.
- Chapter 3, “Configuring SS7/IP Networks,” describes how to configure the components related to communicating across SS7 networks or across SS7 over IP (SIGTRAN) networks.
- Chapter 4, “Configuring the Steel-Belted Radius Files,” describes how to configure the Steel-Belted Radius files that require specific SIM Server configuration.

- Chapter 5, “Special Attribute Handling,” describes configuration tasks for adding location information to Access-Requests, assigning addresses based on access point name, and adding attributes to Access-Accepts.
- Chapter 6, “Configuring the Call Detail Record Accounting Module,” describes how to configure settings specific to SIMAuth in Steel-Belted Radius/SIM Server
- Chapter 7, “Configuring EAP-SIM/EAP-AKA Authentication,” describes configuration tasks for using EAP/SIM or EAP/AKA credentials to authenticate mobile subscribers for wireless hotspot Internet access.
- Chapter 8, “Configuring Web-Based Authentication with SMS,” describes how to configure settings specific to SMS functions in Steel-Belted Radius/SIM Server.
- Appendix A, “SS7 Sample Configuration Files,” provides brief explanations for terminology used in this and other Steel-Belted Radius manuals.
- Appendix B, “Reloading the Configuration with HUP Signals,” lists the configuration settings for which you can execute a `kill-HUP` pid command after you change a configuration setting in Steel-Belted Radius in order to make the configuration change effective.
- Appendix C, “SNMP Traps,” describes the SNMP traps that have been added to the basic SNMP functionality in Steel-Belted Radius to support Steel-Belted Radius/SIM Server.
- Appendix D, “Internal LDAP Directory,” describes the internal LDAP directory to extend session information and store temporary records used for web-based authentication with SMS.
- Appendix E, “Kineto INC/SIM Server S1 Interface,” describes the SIM Server S1 interface with the Kineto INC (IP Network Controller).
- Appendix F, “Glossary,” provides a glossary of terms and acronyms.

Conventions

Table 1 defines text conventions used in this guide.

Table 1: Text Conventions

Convention	Description	Examples
Bold typeface	Indicates buttons, field names, dialog names, and other user interface elements.	Select Enable Custom Instructions on the Host Checker Policy page.
Plain sans serif typeface	Represents: <ul style="list-style-type: none"> ■ Code, commands, and keywords ■ URLs, file names, and directories 	Examples: <ul style="list-style-type: none"> ■ Code: certAttr.OU = 'Retail Products Group' ■ URL: Download the JRE application from: http://java.sun.com/j2se/

Table 1: Text Conventions (continued)

Convention	Description	Examples
<i>Italics</i>	Identifies: <ul style="list-style-type: none"> ■ Terms defined in text ■ Variable elements ■ Book names 	Examples: <ul style="list-style-type: none"> ■ Defined term: A <i>Call Detail Record</i> is a call transaction record that tracks network resources. ■ Variable element: The <i>realm.pro</i> file specifies the control point plug-in. ■ Book name: See the <i>Steel-Belted Radius Administration Guide</i>.

Related Documentation

You are provided with two CDs with Steel-Belted Radius/SIM Server. One CD contains Signalware packages and documentation. The other CD contains the Steel-Belted Radius (including SIM Server) software, a complete set of Steel-Belted Radius documentation, and this *SIM Server Administration Guide*.

The following documents supplement the information in this manual. You can find the following manuals on one of the two CDs provided with Steel-Belted Radius/SIM Server.

- The *Steel-Belted Radius Getting Started Guide* describes how to install, configure, and administer the Steel-Belted Radius software on a server.
- The *Steel-Belted Radius Administration Guide* describes how to configure and administer the Steel-Belted Radius server software.
- The *Steel-Belted Radius Reference Guide* describes the configuration files and settings used by Steel-Belted Radius.
- The *Steel-Belted Radius LDAP Scripting Guide* describes how to use scripts written in the JavaScript programming language to enhance the search capabilities of the Steel-Belted Radius LDAP Authentication module.
- The *ss7readme.txt* file on the CD contains the latest information about features, changes, known problems, and resolved problems.

Ulticom Documentation

You can find the following manuals on one of the two CDs provided with Steel-Belted Radius/SIM Server:

- The *Signalware Architecture Overview Manual* describes the features and architecture of the Signalware platform.
- The *Signalware Solaris Installation Manual* describes how to install and configure the Signalware platform.
- The *Signalware Operator's Reference Manual* describes how to operate and troubleshoot Signalware after it has been installed and configured.

- The *Signalware Board Manual* describes the various SS7 interface boards that are distributed by Ulticom.

Also available on the CD are release notes, compliance tables, errata, glossary, references, release notes, and manual pages.

The Signalware software includes a library of man (manual) pages. After you install the Signalware software, you can review the Signalware man pages by typing *man function* at the Solaris command prompt.

(Example: **\$ man BACKUP-NODE**) These man pages are also available on the documentation CD.

IETF Documentation

Steel-Belted Radius/SIM Server supports the IETF draft (RFC 4186) for EAP-SIM. For more information about the IETF standard, see <http://www.ietf.org/rfc/rfc4186.txt> .

Steel-Belted Radius/SIM Server supports the IETF draft (RFC 4187) for EAP-AKA. For more information about the IETF standard, see <http://www.ietf.org/rfc/rfc4187.txt> .

Contacting Technical Support

For technical support, contact Juniper Networks at support@juniper.net, or at 1-888-314-JTAC (in the United States) or 408-745-9500 (outside the United States).

Check our Web site (<http://www.juniper.net>) for additional information and technical notes. When you are running SBR Administrator, you can select **Web > Steel-Belted Radius User Page** to access a special home page for Steel-Belted Radius users.

When you call technical support, please have the following information at hand:

- Your Steel-Belted Radius product edition and release number (for example, Service Provider Edition version 5.4).
- Information about the server configuration and operating system, including any OS patches that have been applied.
- For licensed products under a current maintenance agreement, your license or support contract number.
- Question or description of the problem, with as much detail as possible.
- Any documentation that might help in resolving the problem, such as error messages, memory dumps, compiler listings, and error logs.

Chapter 1

Overview

Steel-Belted Radius/SIM Server enables GSM and UMTS mobile service providers to offer wireless network access to subscribers through hotspot concession operators while leveraging existing customer care, roaming, and billing infrastructures.

Steel-Belted Radius/SIM Server provides two complementary solutions to provide wireless LAN access to subscribers:

- SIMAuth support uses the provider SS7 infrastructure to facilitate secure 802.1X-based subscriber authentication and billing.
- SMS support uses the provider SS7 infrastructure to facilitate subscriber time-limited account provisioning, authentication, and billing.

Steel-Belted Radius/SIM Server Components

This section describes the hardware and software components of Steel-Belted Radius/SIM Server.

Table 2 on page 17 summarizes system requirements.

Hardware Components

Steel-Belted Radius/SIM Server requires a Sun server running Solaris 9 and one or more Ulticom SS7 line interface boards. (The board is not necessary if you use only SIGTRAN for SS7 over IP.)

Server

Steel-Belted Radius/SIM Server runs on a Sun Server with at least 1 Gb of memory. If you are using Signalware for SS7 communication, you must have a slot in the server for installing a PCI board. Steel-Belted Radius/SIM Server 5.4 is compatible only with Sun Solaris 9.

Ulticom SS7 Interface Board

The basic configuration of Steel-Belted Radius/SIM Server includes one Ulticom T1/E1 interface board that supports a specified number of 56k/64k links. The interface board plugs into an available PCI slot in your server. You can configure this board to support T1 (North American) or E1 (European) electrical characteristics by setting jumpers on the board. Refer to the Ulticom documentation (provided with SIM Server) for information about installing and configuring your Ulticom SS7 interface board.

Software Components

Steel-Belted Radius/SIM Server includes the following software components:

- Steel-Belted Radius/Service Provider Edition software
- Ulticom Signalware SS7 protocol stack
- SMS provisioning and authentication modules
- CDR accounting module
- SIM/AKA authentication (SIMAuth) module
- MAP (Mobile Application Part) Gateway
- SMS Gateway
- Database accessors

Steel-Belted Radius/Service Provider Edition

The Steel-Belted Radius/SPE software provides RADIUS AAA (authentication, authorization, and accounting) services that handle access requests from your IP network.

- For SMS support in Steel-Belted Radius/SIM Server, a service provider's Access Controller (AC) generates authentication requests to the Steel-Belted Radius server.
- For SIMAuth support in Steel-Belted Radius/SIM Server, a service provider's 802.1X Access Point (AP) directs authentication requests from user devices to the Steel-Belted Radius server.

Ulticom Signalware SS7 and/or SIGTRAN Protocol Stack(s)

The SS7 protocol stack handles the various SS7 protocol layers to put the MAP requests onto the SS7 network. ANSI SS7, CCITT/ITU SS7, Japanese, and Chinese networks are supported. SIGTRAN provides SS7 signaling over IP networks.

SMS Provisioning and Authentication Modules

Steel-Belted Radius/SIM Server includes SMS provisioning and authentication modules.

- The SMS provisioning module handles all account creation requests that come to it from Access Controllers. The SMS provisioning module also generates the one-time password that is sent to the user through an SMS text message. Depending on a service provider's configuration, the SMS provisioning module interacts with an LDAP (Lightweight Directory Access Protocol) or SQL database or with the HLR to check that the request is for a subscriber belonging to the operator and that the subscriber is authorized for WLAN access.
- The SMS authentication module authenticates access requests against the provisioned user accounts.

CDR Accounting Module

Steel-Belted Radius/SIM Server includes a Call Detail Record (CDR) accounting module. The CDR module manages all CDR-based subscriber accounting for the purposes of billing. CDRs are forwarded through FTP server or other transport method to a billing application.

SIM Authentication Module

Steel-Belted Radius/SIM Server includes an EAP-SIM/EAP-AKA (SIMAuth) authentication module. The SIMAuth module manages all EAP-SIM-based and EAP-AKA based authentication requests from subscribers.

- EAP-SIM authentication is used with older SIM cards.
- EAP-AKA authentication is used with third-generation USIM (Universal Subscriber Identity Module) cards.

The SIMAuth module supports user authentication based on SIM IMSI (International Mobile Subscriber Identity) values or on anonymous authentication based on pseudonym values that are assigned after the first successful authentication.

MAP Gateway

The MAP Gateway acts as a link between Steel-Belted Radius and the SS7 network. It formats and transmits MAP requests to the HLR over the SS7 network. The MAP Gateway processes requests for authentication and authorization information.

SMS Gateway

The SMS Gateway acts as a link between Steel-Belted Radius and the SS7 network. It formats and transmits MAP requests to the MSC (Mobile Switching Center) over the SS7 network. The SMS Gateway processes requests for MSISDN-to-IMSI mappings and sends SMS messages to subscribers' cell phones or mobile devices.

Database Accessors

Database accessors enable Steel-Belted Radius/SIM Server to query an external SQL database or LDAP directory server for WLAN authorization and IMSI/MSISDN lookups. Database accessors can be used to supplement or replace the interaction between Steel-Belted Radius/SIM Server and a service provider's HLR.

You must install an Oracle 7,8, or 9 client or JDBC (Java Database Connectivity) if you want to use the SQL data accessor.

Web-Based Authentication Using SMS

Steel-Belted Radius/SIM Server provides mobile service providers the ability to bill mobile subscribers for wireless hotspot Internet access. SMS (short message service) support in Steel-Belted Radius/SIM Server requires that mobile subscribers have the following equipment present when they request service:

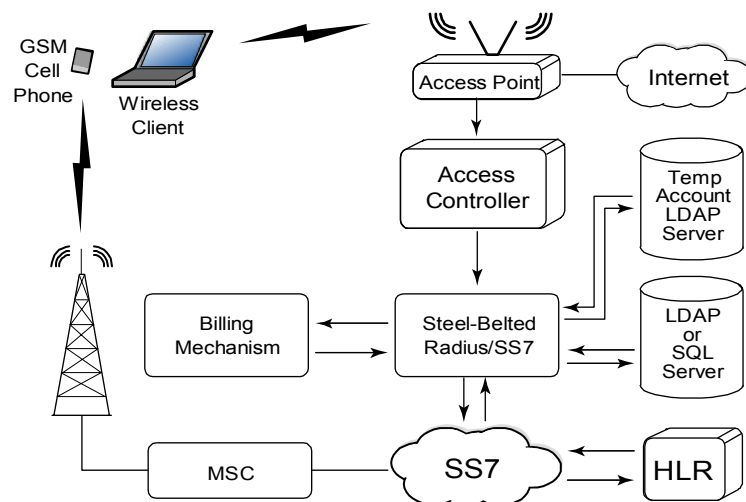
- A mobile device, such as a cellular telephone
- A wireless client, such as a laptop computer or personal digital assistant (PDA) with a wireless network adapter

SMS offers a secure method for password-provisioned public WLAN access through a RADIUS/IP-based public network. The subscriber's temporary password is transmitted securely to the subscriber's device using SMS. You can configure Steel-Belted Radius/SIM Server so that the transmitted message is composed in a language of choice of the subscriber.

This solution does not provide the security inherent in using 802.1X, but in the absence of 802.1X-enabled hotspots and/or 802.1X-capable devices, it provides a means for subscribers to be authenticated to open WiFi networks to obtain network access, where they are directed to a Web page that forces them to enter user name and password.

Figure 1 presents an overview of the SMS access solution.

Figure 1: SMS Support in Steel-Belted Radius/SIM Server



The SMS access solution functions as follows:

1. A subscriber equipped with a wireless-enabled laptop or PDA and a cell phone connects to a hotspot Access Point.
 - The Access Point assigns an IP address to the subscriber.
 - When the subscriber tries to access the Internet, the Access Controller redirects the subscriber to a forced landing page (presented by a standalone or integrated Web server) that asks for a telephone number to which a temporary password can be sent.
2. When the subscriber enters a telephone number (MSISDN), Steel-Belted Radius/SIM Server uses the service provider's subscriber management system to authenticate the subscriber and to create billing records. Steel-Belted Radius can use an external LDAP or SQL database, or an HLR on the SS7 network to perform subscriber authorization.
3. If the user is authorized for WLAN access, Steel-Belted Radius/SIM Server generates a random password and provisions a time-limited Internet access account.
4. Steel-Belted Radius/SIM Server returns the account password to the subscriber cell phone in an SMS text message transmitted through the provider mobile switching center (MSC). The provider can configure Steel-Belted Radius to offer a choice of languages for text messages to users requesting wireless LAN access.
5. The subscriber uses the temporary password included with the SMS text message, along with the subscriber's telephone number (MSISDN), to request access to the public hotspot through the redirected web page.
6. When the subscriber is authenticated for the provisioned account, the Access Controller sends an accounting message to the Steel-Belted Radius/SIM Server, and the subscriber is charged a connection fee.

SMS allows a subscriber to be authenticated for access to an IP network. The subscriber obtains the required password from a message that is sent to the subscriber phone.

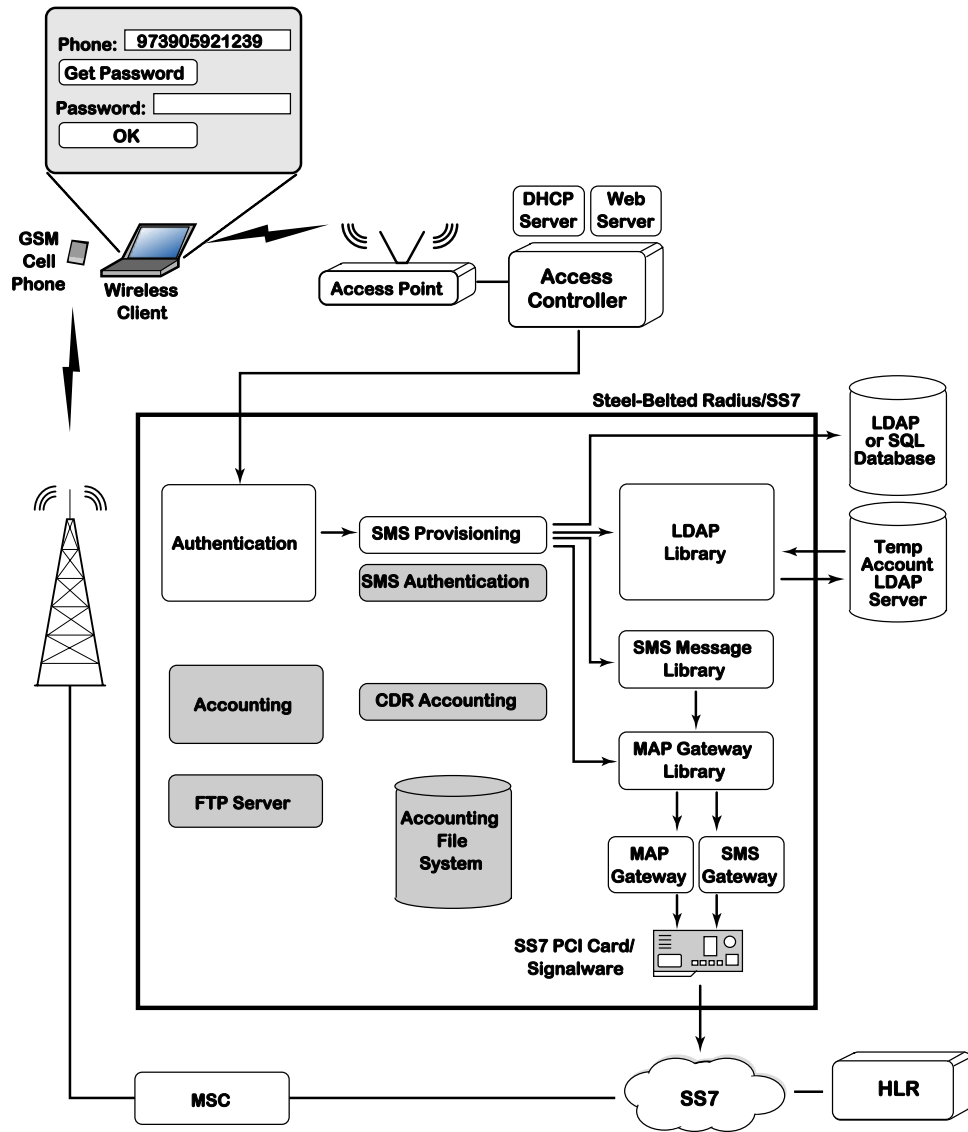
The Steel-Belted Radius/SIM Server solution includes the following three distinct processes/features:

- Authorization and temporary account provisioning
- Authentication
- Accounting and billing

SMS Authorization and Account Provisioning

Figure 2 shows the SMS authorization and account provisioning process.

Figure 2: SMS Authorization and Provisioning



Account authorization and provisioning begins with an account request by a subscriber, and ends with return of a temporary password in the form of a text message to the subscriber cell phone.

The following sequence takes place when a wireless client requests the creation of a temporary account to an IP network protected by Steel-Belted Radius/SIM Server:

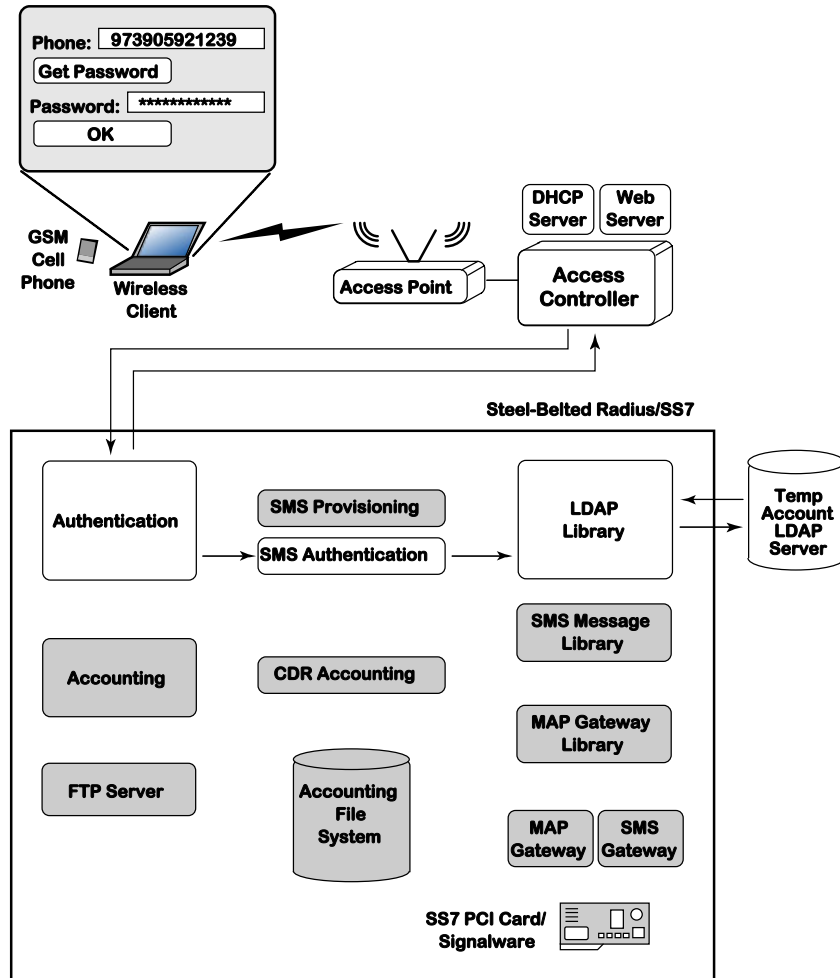
1. A subscriber equipped with a cell phone and a wireless computer or PDA requests an Internet connection at a WLAN hotspot.

2. The subscriber is granted access to the Access Point, and is provided a local IP address by a DHCP (Dynamic Host Configuration Protocol) server located behind the Access Point.
3. The Access Point directs the subscriber to a registration Web page. The subscriber is prevented from using protocols other than HTTP or HTTPS. The Web page, which can be configured to display messages in different languages, prompts the subscriber to enter the phone number (MSISDN) of his or her cell phone.
4. Steel-Belted Radius/SIM Server checks the temporary account table to see if a provisioned account for the subscriber exists:
 - If an unexpired account for the subscriber exists, Steel-Belted Radius/SIM Server checks whether the subscriber has repeatedly attempted to log in unsuccessfully (which might indicate an unauthorized user). If the number of user login attempts is within an acceptable range, Steel-Belted Radius/SIM Server formats a password reminder message and sends the message to the subscriber's cell phone.
 - If an unexpired account for the subscriber does not exist, the SMS provisioning module provides the MAP Gateway with the subscriber MSISDN. The MAP Gateway retrieves the subscriber IMSI and queries the appropriate HLR (or an external SQL database or LDAP directory) to determine whether the subscriber is authorized for WLAN access.
 - If the subscriber is authorized for WLAN access, the SMS provisioning module creates a temporary account (using the subscriber MSISDN as the account name) and generates an account password. The SMS provisioning module forwards a text message containing the account password to the SMS Gateway, which forwards the message to the MSC, which sends the message to the subscriber's cell phone.
 - If the subscriber is not authorized for WLAN access, Steel-Belted Radius/SIM Server forwards an error message to the access controller and terminates SMS provisioning.
5. Steel-Belted Radius/SIM Server returns a RADIUS Access-Reject message to the Access Point to indicate that the account has been provisioned but the user has not been authenticated yet. (An Access-Accept message will be generated after Steel-Belted Radius/SIM Server authenticates the user.)

SMS Authentication

Figure 3 shows the SMS authentication process.

Figure 3: SMS Authentication



After Steel-Belted Radius/SIM Server confirms that a user is authorized to access the WLAN and provisions a temporary account, it authenticates the user as follows:

1. The subscriber enters his or her telephone number (MSISDN) and the password associated with his or her provisioned account.
2. The Access Controller sends a RADIUS authentication request identifying the requestor as the subscriber MSISDN and realm for the provider. The Access Controller encrypts the password that the subscriber entered in the registration page and includes it in the access request.
3. Steel-Belted Radius/SIM Server passes the authentication request to the SMS authentication module (smsauth). The SMS authentication module queries the temporary account directory, retrieves the subscriber account information (including the password provisioned for the account), and verifies that the password in the access request is correct.

4. Steel-Belted Radius/SIM Server sends a RADIUS Access-Accept response to the Access Controller and configures an account expiration time.

After the Access Point receives the Access-Accept message and grants WLAN access to the subscriber, the subscriber can close and open sessions using the same temporary password until the account expires. When the temporary account expires, the subscriber must request another provisioned account to obtain continued network access.

The temporary account directory retains information about the account after the account expires so that Steel-Belted Radius/SIM Server can distinguish between a nonexistent account and a recently-expired account. An account sweeper periodically deletes expired accounts that are more than a certain amount of time past their expiration.

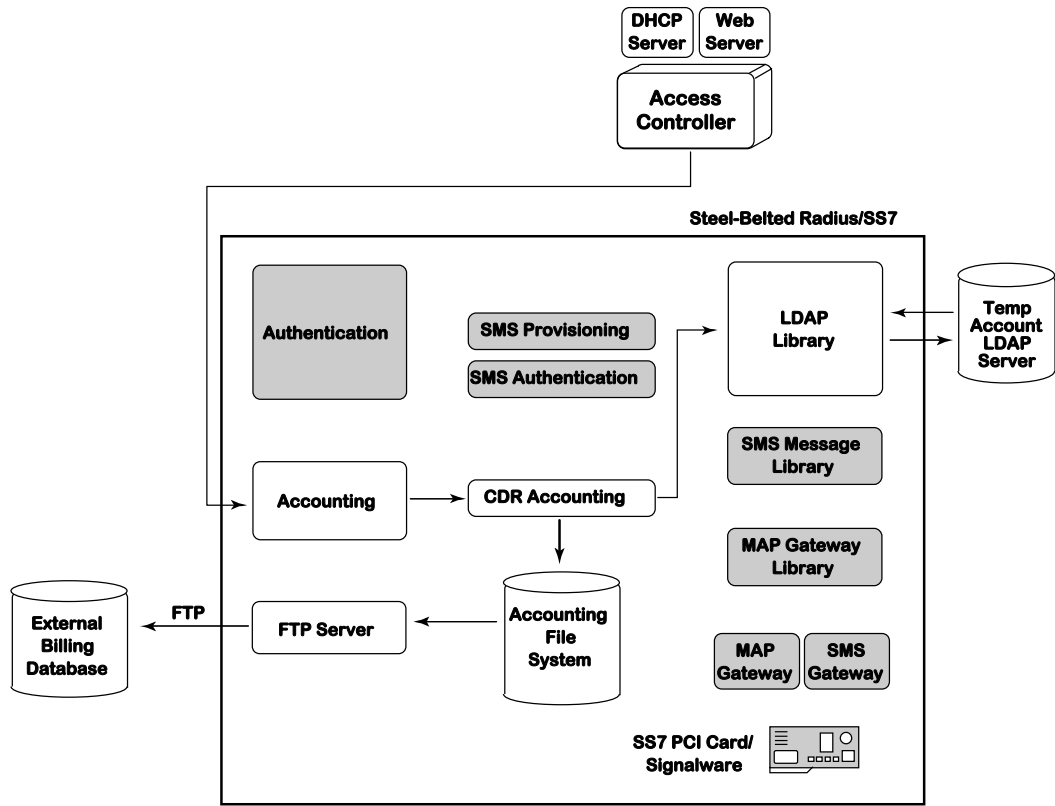
CDR Accounting and Billing

Steel-Belted Radius/SIM Server includes a Call Detail Record (CDR) accounting module. The CDR module manages all CDR-based subscriber accounting for the purposes of billing. CDRs are forwarded to a charging gateway through an FTP server or other transport method to a billing application.

Figure 4 shows the call detail record (CDR) accounting process.

See “Configuring the Call Detail Record Accounting Module” on page 105 for more information about the CDR module,

Figure 4: CDR Accounting



Fraud Prevention

Steel-Belted Radius/SIM Server can be configured to identify situations where more than one user is logged in at the same time using the same MSISDN/password combination. When a user attempts to log in, the remote Access Controller (AC) forwards the MAC address of the user’s computer in the Calling-Station-Id attribute of the access request. The Steel-Belted Radius server stores the user’s MAC address in the temporary account for that MSISDN. Thereafter, subsequent authentication requests from that MSISDN must originate from the MAC address matching the address stored in the temporary account directory.

The remote AC must support forwarding of a user’s MAC address in the Calling-Station-Id attribute of an Access Request message for this feature to work. If the remote AC does not forward the MAC address for a user, MAC address checking is disabled for that user.

SIMAuth Support

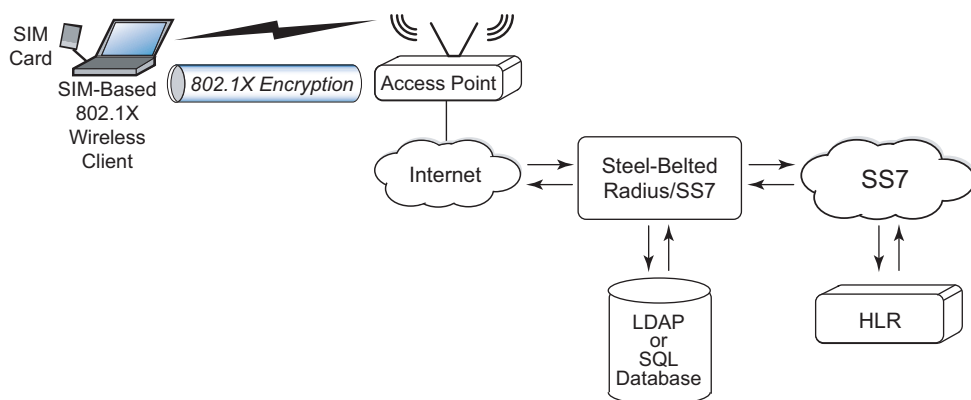
SIMAuth offers secure public WLAN access. Subscriber data is transmitted securely over the wireless link. Data security is ensured with either the Wi-Fi Protected Access (WPA) or the dynamic WEP encryption protocol to prevent wireless eavesdropping within the hotspot.

SIMAuth serves as the link between the IP network and the SS7 network, verifying SIM-based or USIM-based credentials provided by a subscriber against an existing provider Home Locator Register (HLR), SQL database, or LDAP directory, to authenticate the subscriber and obtain authorization information.

SIMAuth also provides the ability to bill mobile subscribers using the Call Detail Record (CDR) accounting module.

To access a public hotspot, the subscriber's wireless laptop or PDA must include an IEEE 802.1X supplicant (client application) that supports EAP-SIM or EAP-AKA, such as Juniper Network's Odyssey Client software.

Figure 5: SIMAuth Support in Steel-Belted Radius/SIM Server



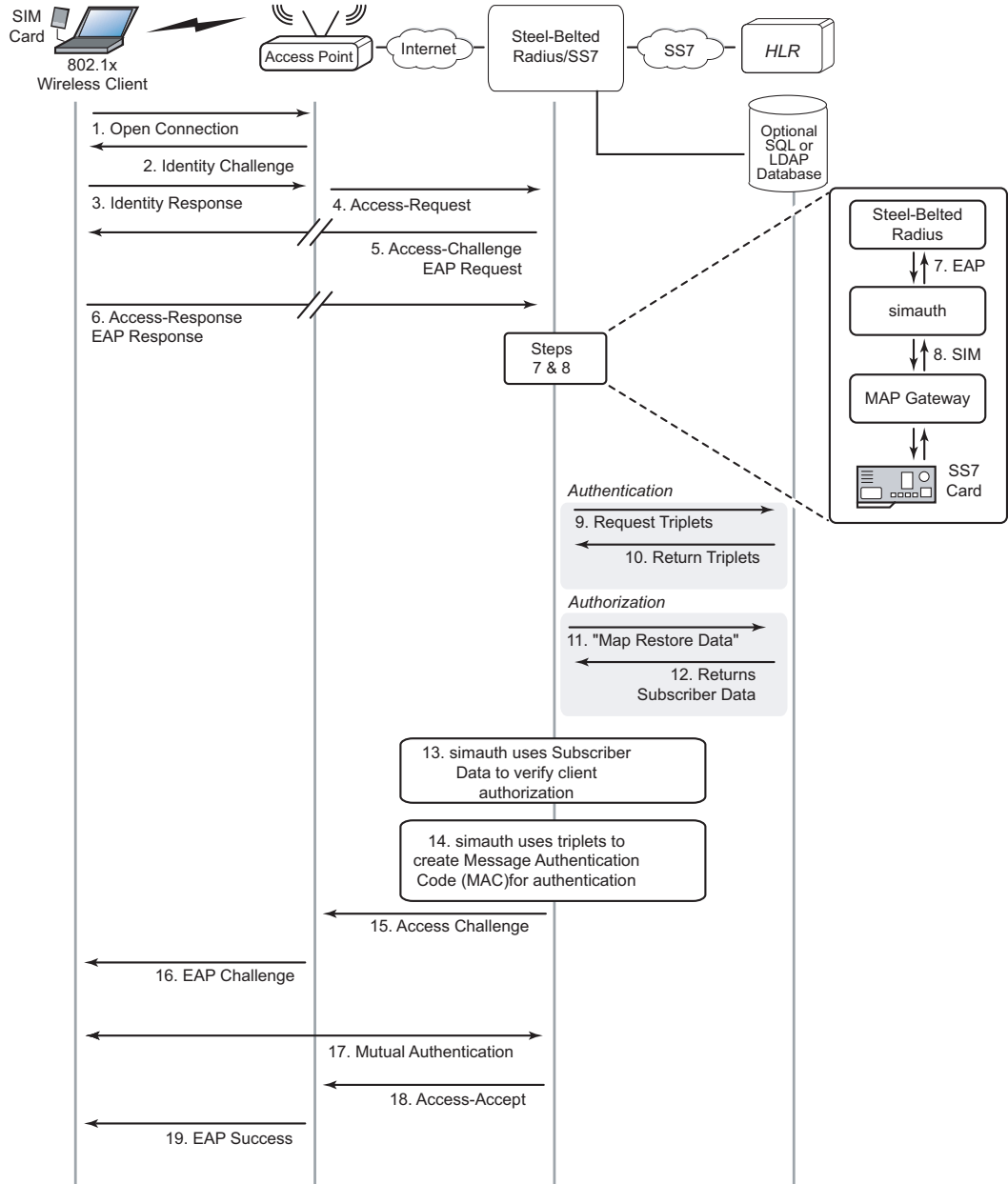
Steel-Belted Radius/SIM Server supports EAP-SIM/EAP-AKA fast reauthentication. As a result, your server can regularly replace the encryption keys used over the wireless link to provide identity protection to subscribers. Reauthentication occurs securely without requiring interaction between Steel-Belted Radius/SIM Server and the HLR, thereby decreasing traffic load on the RADIUS server.

To perform an EAP-SIM or EAP-AKA authentication, the wireless Access Point must have an EAP-capable 802.1X supplicant. EAP-SIM can be used with HLRs that support MAP application context version 2 and EAP-AKA can be used with HLRs that support version 3.

SIM Card-Based Authentication

Figure 6 shows the SIM-based provider/subscriber solution supported by Steel-Belted Radius/SIM Server.

Figure 6: Steel-Belted Radius/SIM Server Authentication



The following sequence takes place when a wireless client with a SIM card installed requests access to an IP network protected by Steel-Belted Radius/SIM Server:

1. A user running Odyssey Client or other EAP-SIM-compatible 802.1X supplicant opens a wireless connection to an 802.1X Access Point at a WLAN hotspot.
2. The 802.1X-configured Access Point challenges the supplicant for its identity.
3. The 802.1X supplicant on the wireless client responds with an EAP-SIM-based identity.
4. The Access Point forwards this information to Steel-Belted Radius/SIM Server (directly or through a proxy RADIUS server).
5. Steel-Belted Radius/SIM Server sends an Access-Challenge request for EAP-SIM authentication to the Access Point, which forwards the request to the wireless client.
6. The wireless client receives the EAP-SIM request, and, through the Access Point, agrees to start EAP-SIM by sending an EAP-SIM response. Included in the wireless client response is a *nonce* (large random number) that protects against playback attacks.
7. The Steel-Belted Radius software passes the EAP information to the **simauth** module.
8. The **simauth** module converts the EAP information to a SIM request and passes it to the MAP Gateway.
9. The MAP Gateway passes a request for information that it needs to perform the authentication (called *triplets*) through the SS7 network to the HLR.
10. The HLR does a database lookup and returns the requested triplets to the **simauth** module.
11. If Steel-Belted Radius/SIM Server is configured to support user authorization, the **simauth** module sends a request for authorization information to an external (LDAP or SQL) database or through the SS7 network to the HLR.
12. The external database or HLR returns the requested authorization information for the user.
13. The **simauth** module uses subscriber data to verify authorization.
14. The **simauth** module uses these triplets to create the message authentication code (MAC) that it sends as part of its challenge to the wireless client.
15. Steel-Belted Radius/SIM Server sends an Access-Challenge containing the MAC to the Access Point.
16. The Access Point forwards the challenge to the wireless client.
17. The wireless client verifies that the message is authentic (by running the appropriate authentication algorithms) and responds by sending its own message authentication code to Steel-Belted Radius.

18. The Steel-Belted Radius server verifies the client message authentication code, and sends an Access-Accept response to the wireless Access Point. The Access-Accept includes keying material for encrypting data sent on the wireless connection.
19. The wireless Access Point uses the keying material from the Access-Accept to establish an encrypted session with the wireless client.

Authentication using a USIM card and EAP-AKA is similar, although the authentication information retrieved from the HLR is different.

EAP-SIM/EAP-AKA Authorization/Service Delivery

In addition to authenticating SIM-based users to the public network and setting up their secure connections, Steel-Belted Radius/SIM Server lets you configure subscriber connections according to authorization strings that you can configure in the subscriber database or HLR. With Steel-Belted Radius/SIM Server, you can map one or more authorization strings to a profile that you configure in Steel-Belted Radius. This Steel-Belted Radius profile is applied to the user connection.

EAP-SIM/EAP-AKA Identities

To provide identity protection and fast reauthentication, the EAP-SIM/EAP-AKA protocol includes three types of identities for the client. For any given authentication between the client and server, only one of these three is required:

- The *Permanent Identity* is the identity required by the EAP-SIM/EAP-AKA server to retrieve the authentication information (triplets) and authorization information from the external database or HLR. The Permanent Identity must contain the IMSI from the SIM card being used.
- The *Pseudonym Identity* is used in place of the Permanent Identity whenever it is available. The EAP-SIM/EAP-AKA server (Steel-Belted Radius/SIM Server) creates the pseudonym, and sends it to the client on the first authentication. A new pseudonym is created every time that authentication occurs. Steel-Belted Radius/SIM Server uses an encrypted form of the IMSI as the pseudonym so that pseudonyms can be shared among all Steel-Belted Radius servers that share the same encryption key. The Pseudonym Identity provides identity protection by hiding the Permanent Identity (the IMSI) on the second and all future authentications.
- The *Reauthentication Identity* also provides identity protection. Like the pseudonym, a new Reauthentication Identity is created by the server on each authentication. However, the purpose of using the Reauthentication Identity is to begin a fast reauthentication exchange. During this exchange in EAP-SIM or EAP-AKA, new key material is generated based on the current authentication information. The SIM card and the HLR do not need to participate in this exchange. This feature is used to regularly replace the encryption keys used over the wireless link.

You can disable the Pseudonym and Reauthentication identities. See “simauth.aut [Settings] Section” on page 124 for details.

Any of these identities can also contain an *@realm* tag to form a Network Address Identifier (NAI). The wireless Access Point typically uses the EAP identity as the RADIUS username for communicating to the RADIUS server. In this case, the RADIUS server or RADIUS proxy server can use the realm to direct the RADIUS request to another server or use the realm in another way.

The EAP-SIM and EAP-AKA protocols allow the server to provide a realm when it creates a Reauthentication Identity. Steel-Belted Radius/SIM Server typically uses the realm that was received with the last Permanent Identity or Pseudonym Identity for the realm returned with the new Reauthentication Identity. You can configure a different realm to be returned in the `simauth.aut` configuration file.

EAP-SIM/EAP-AKA Fast Reauthentication

You can configure Steel-Belted Radius/SIM Server for fast reauthentication, so that once a customer establishes a secure 802.1X network connection through the authentication and authorization processes, the server replaces the encryption keys used over the wireless link. Since fast reauthentication does not require interaction between Steel-Belted Radius/SIM Server and the HLR, this added security measure does not affect traffic loads on the SS7 network.

See “`simauth.aut` [Settings] Section” on page 124 for information about configuring fast reauthentication features.

Chapter 2

System Setup

This chapter describes how to set up the Steel-Belted Radius/SIM Server software and related components. Refer to the Steel-Belted Radius and Ulticom Signalware documentation (provided with the software) for additional information.

System Requirements

Table 2 lists the system requirements for running the Steel-Belted Radius/SIM Server software.

Table 2: System Requirements

Component	Requirement
Operating system	Solaris 9 with patch cluster newer than April 2005 with the 64-bit packages installed.
Memory	At least 1 GB of memory
Disk space	At least 1000 MB for files and execution
SS7 board	Ulticom-approved SS7 board, such as PCI Bus SS7 T1/E1 Interface Board - PH0301, which is shipped with SIM Server (if ordered). (Needed only if using SS7 network.)

Installing and Running the Configuration Script

Steel-Belted Radius/SIM Server is provided as a single, integrated software product. Your installation of Steel-Belted Radius includes SIM Server if you are licensed for SIM Server.



NOTE: If you already have a previous version of SIM Server and want to upgrade to the most current version, see “Upgrading Solaris, Signalware, and SIM Server” on page 35.

To install and run the configuration script for Steel-Belted Radius/SIM Server on a Solaris server, follow the steps described in the following procedures:

- “Installing Steel-Belted Radius/SIM Server” on page 18
- “Running the Configuration Script” on page 19

Installing Steel-Belted Radius/SIM Server

The installation procedure copies the installation files to the Solaris server and runs the installer package.

To install Steel-Belted Radius/SIM Server:

1. Log into the Solaris server as root.
2. Copy the Steel-Belted Radius installation files to the Solaris server.

Copy the files from the `/solaris` directory on the installation CD to a local or remote hard disk partition that is readable by `root`.

The following example copies the files to the `/opt/juniper/temp` directory.

```
# mkdir -p /opt/juniper/temp
# cp -pR /cdrom/solaris/* /opt/juniper/temp
```

If extraneous error messages similar to the following appear when you insert the installation CD-ROM, ignore them.

```
rigel does not conform to the ISO-9660 specification:
rigel hsfs: file len greater than max allowed
rigel hsfs: Due to this error, the file system may not be correctly interpreted.
rigel hsfs: Other such errors in this file system will be silently ignored.
```

3. Change to the directory into which you copied the files in step 2.
4. Uncompress the Steel-Belted Radius installation package using the following commands.

```
# gunzip -d JNPRsbrsim.pkg.tgz
# tar -oxf JNPRsbrsim.pkg.tar
```

5. Run `pkgadd`.

```
# pkgadd -d directory JNPRsbrsim.pkg
```

where *directory* specifies the directory where you placed the package.

Example:

In the following command, the directory that follows the `-d` parameter specifies the location where the `JNPRsbrsim.pkg` can be found. In this example (in step 2 above), the package was copied to `/opt/juniper/temp` and that is the location where it can be found.

```
# pkgadd -d /opt/juniper/temp JNPRsbrsim.pkg
```



NOTE: By default, the installation package puts the Steel-Belted Radius files in the `/opt/funk/radius` directory.

6. When you are prompted to confirm that you want to install the package, enter `y`.

Do you want to continue with the installation of JNPRsbrsim [y,n,?] y

7. Proceed to the next procedure, “Running the Configuration Script.”

Running the Configuration Script

The configuration script for Steel-Belted Radius/SIM Server enables you to enter basic configuration choices.

To run the configuration script for Steel-Belted Radius/SIM Server:

1. Navigate to the directory where you installed Steel-Belted Radius.

```
# cd /opt/funk/radius/install
```

2. Execute the following command to run the configuration script for the Steel-Belted Radius server software:

```
# ./configure
```

3. Review the Steel-Belted Radius license agreement.
4. Press the spacebar to move from one page to the next. When you are prompted to accept the terms of the license agreement, enter Y.

```
Do you accept the terms in the license agreement? [y]
```

5. Indicate that you have a license number.

```
Would you like to enter a license string? [y]
```

6. When prompted to do so, enter your license number and press **Enter**. (Your license number can be found on a sticker affixed to the license agreement in your product package or check with your Juniper representative.) The script creates your license file and copies it to your server directory.
7. Specify whether you want to install the Steel-Belted Radius server as a primary server (p), a replica server (r), or a standalone RADIUS server (sa).

```
Configure SBR server as primary (p), replica (r), or standalone (sa) [sa]: sa
```

- If you enter p (primary server), you are prompted to enter the replication secret used to authenticate communications between the primary server and replica servers. Enter and confirm the replication secret and press **Enter** to continue.

If appropriate, enter y when you are prompted whether you are upgrading a primary server. Doing so tells the installer to preserve the server’s replication realm information.

- If you enter r (replica server), you are prompted to specify how the replica server can locate the `replica.ccmpkg` configuration package containing your Steel-Belted Radius replication settings.

- ❑ If the replication package is present on your computer or network, you are prompted to specify the path to the `replica.ccmpkg` file.
 - ❑ If you want to specify the location of the primary server (from which the replica server can copy its replication package automatically), enter the name, IP address(es), and replication secret of the primary server.
 - ❑ If appropriate, enter `y` when you are asked whether you are upgrading a replica server. Doing so tells the installer to preserve the replica server's replication settings.
 - If you enter `sa` (standalone RADIUS server), you do not need to specify replication information.
8. Specify the login name of the initial Steel-Belted Radius administrator. The account information that you enter is the default login account for the SBR Administrator. You must use this account name the first time that you log into the SBR Administrator.

Enter initial admin user (account must have an associated password) [root]:



NOTE: Make sure that the login account that you specify has a password. If a user without a password is specified as the administrator, the user will not be able to log into the SBR Administrator.

9. Specify whether you want to configure Steel-Belted Radius for use with an external LDAP data service.

Do you want to configure LDAP? [n]:

If no, press **Enter**. Steel-Belted Radius is configured to use a generic database.

If yes, type `Y` and press **Enter**. You are prompted to enter the path for the LDAP library files:

Enter path for LDAP library files [/usr/lib]:

To accept the default path (`/usr/lib`), press **Enter**.

10. Specify whether you want to configure Steel-Belted Radius for use with an External SQL database or an Oracle database.

Do you want to configure for use with External SQL Databases? [n]:

If no, press **Enter** and continue with step 11 to configure SNMP.

If yes, type `Y` and press **Enter**.

Do you want to configure for use with Oracle? [n]:

If no, press **Enter** to use JDBC and continue with step 11 to configure SNMP.

If yes, type `Y` and press **Enter**. You are prompted to enter the path for the Oracle library files.

Configuring for use with Oracle.
 Supported Oracle version: 8, 9
 What version of Oracle will be used? 9
 Configuring for use with Oracle 9.
 Setting the environment variable ORACLE_HOME.
 Enter ORACLE_HOME:
 Setting the environment variable LD_LIBRARY_PATH.
 Enter path for Oracle shared libraries:
 Setting the environment variable TNS_ADMIN.
 Enter TNS_ADMIN:

11. Specify whether you want to install the optional SNMP module so that you can monitor your Steel-Belted Radius server from an SNMP management station.

Do you want to configure SNMP? [n]:

If no, press **Enter** to proceed to the next prompt.

If yes, type Y and press **Enter**. The installer prompts you for information.

- When you are prompted for a community string, enter the community string used to validate information sent from the SNMP subagent on the Steel-Belted Radius server to your SNMP management station.

Choose a community string: public

- When you are prompted for a range of IPv4 addresses, specify a starting IP address in Classless Inter-Domain Routing format. To specify that only one host can query the agent, enter the IP address of the host followed by /32. To specify that any host on a designated class C network can query the agent, enter the starting address of the network followed by /24.

Specify the range of IPv4 addresses that may query this agent, such as 1.2.3.0/24.

Address range:

See the *Steel-Belted Radius Getting Started* for more information about using CIDR notation to specify IP address ranges.

- If you are using SNMPv2, enter the DNS name or IP address of the trap sink that will receive trap information from the SNMP subagent on the Steel-Belted Radius server.

Optionally specify a trap sink that will receive SNMPv2 traps:
 Configuration of SNMP complete.

Refer to the *Steel-Belted Radius Administration Guide* for configuration information related to the SNMP agent. Refer to the remaining chapters in this manual for specific configuration information for SIM Server.

12. When asked whether or not you want to register your Steel-Belted Radius server as an Agent Host with RSA Authentication Manager, type n.

Do you want register SBR with an RSA server (requires RSA Auth Manager 6.1 or later)? [n]:

When you finish entering settings, the script configures Steel-Belted Radius with the settings that you specified.

Steel-Belted Radius/SIM Server can now be configured to meet your specific requirements. Refer to the *Steel-Belted Radius Administration Guide* for configuration information related to Steel-Belted Radius. Refer to the remaining chapters in this manual for specific configuration information for SIM Server.

Starting and Stopping Steel-Belted Radius/SIM Server

The version of Steel-Belted Radius that includes SIM Server starts the OpenLDAP server at the same time that you start Steel-Belted Radius. There is no need to start OpenLDAP separately.



NOTE: If only OpenLDAP is running or only Steel-Belted Radius is running, you must stop the process that is open and then start Steel-Belted Radius.

Example

Change to the `radius` directory:

```
cd /opt/funk/radius
```

Check to see if Steel-Belted Radius or OpenLDAP is running:

```
./sbrd status
```

Stop OpenLDAP and Steel-Belted Radius if they are already running:

```
./sbrd stop
```

Start both Steel-Belted Radius and OpenLDAP:

```
./sbrd start
```

The `./sbrd restart` command is also valid. The `./sbrd restart` command stops Steel-Belted Radius and OpenLDAP and then starts them again.



NOTE: Steel-Belted Radius shuts down and starts automatically each time that you shut down or restart the server. (This is true only if `watchdog` is enabled. See the *Steel-Belted Radius Reference Guide* for more information about the `watchdog` process.)

Correlating Related Log Entries (Log Thread)

Steel-Belted Radius uses log files to track and report information about Access-Requests. When multiple requests are processed simultaneously, log entries for different requests might appear consecutively in the log file.

You can configure the `radius.ini` file to include a thread identification number with log entries, which correlates the log entries produced while processing each RADIUS request. To cause the thread identification number to be logged, enter the following lines in the `radius.ini` file:

```
[Debug]
Log-Thread-ID=yes
```

In the following example, the Log-Thread-ID of 98 is assigned to one Access-Request and 73 is assigned to another.

```

      :
      :
08/24/2006 15:16:27 ../radauthd.c radAuthHandleRequest() 2720 (98) Entering
08/24/2006 15:16:27 (98) Looking up shared secret
08/24/2006 15:16:27 (98) Looking for RAS client 172.25.97.54 in DB
08/24/2006 15:16:27 (98) Matched 172.25.97.54 to RAS client <ANY>
08/24/2006 15:16:27 (98) Parsing request
08/24/2006 15:16:27 (98) Initializing cache entry
08/24/2006 15:16:27 (98) Doing inventory check on request
08/24/2006 15:16:27 (98) Getting info on requesting client
08/24/2006 15:16:27 (98) User-Name : String Value = 1212864080212345
      :
      :
08/24/2006 15:16:27 (73) Authentication Request
08/24/2006 15:16:27 (73) Received from: ip=172.25.97.54 port=4334
08/24/2006 15:16:27 (73)
08/24/2006 15:16:27 (73) Raw Packet :
      :
      :
```

Installing Signalware 9.0

Signalware provides the ability to communicate across SS7 networks. SIGTRAN (provided as a module of Signalware) enables SS7 communication over IP networks.

If you already have another version of Signalware, refer to “Upgrading Solaris, Signalware, and SIM Server” on page 35 and “Upgrading from Signalware 8 to Signalware 9” on page 38.

This section provides information about installing Signalware 9 for use with SIM Server. The installation procedure provided here is for your convenience and assumes that this is a “clean” (first time) installation of Signalware. For more complete information about installing Signalware (including re-installation of Signalware 9), refer to the *Signalware Solaris Installation Manual* that is provided with the SIM Server software.

To install Signalware 9:

1. If you are upgrading from Signalware 8, see “Upgrading from Signalware 8 to Signalware 9” on page 38.
2. Place the Signalware 9 license file in the `/etc` directory.

To obtain a license file for Signalware 9, contact your Juniper representative. To find out information about your license, type `liceManViewer` at the Solaris prompt.

3. Change to superuser mode.
4. Create a UNIX group called “users” and verify that the group exists.

To add a group, use the UNIX `groupadd` command.
`$ groupadd users`

To view a list of existing groups, use the UNIX `cat` command:
`$ cat /etc/group`

5. Insert the Signalware CD and navigate to the Signalware Packages directory.



NOTE: You are provided two CDs with Steel-Belted Radius/SIM Server. One CD contains Signalware packages and documentation. The other CD contains the Steel-Belted Radius (including SIM Server) software, a complete set of Steel-Belted Radius documentation, and the SIM Server documentation.

6. Start the Signalware installation menu by running `swsetup`, located in the `signalwarePackages` directory.

A request for a user identifier appears.

7. Enter a unique user identifier and press **Enter**. You can use any identifier, provided it is 15 characters or less.

Please Enter a unique User Identifier.
 It cannot be more than 15 characters long
 For example: username-extension (jdoe-1234):my-user-name

8. Enter Y or N when prompted with the following question: Searching for updates to the Signalware UTIL package...Could not find an update...Do you have an ECN Update to the OMNI-UTIL package? Normally, respond with N.

A request for a scheduling priority appears.

9. Accept the default value of 10 and press **Enter**.

Please enter a scheduling priority, between 1 and 19, for all resource intensive processes spawned by the Signalware menu system [10]

The Main Menu screen appears.

10. Enter **1** for Install/Configure and press **Enter**.

```

Main Menu
1 = Install/Configure (Signalware is uninstalled or off-line)
2 = Online Upgrade (Signalware is installed and running)
3 = Installation Status and Reports
4 = Installation Maintenance
5 = Configuration Maintenance
6 = Start an Installed Instance of Signalware
7 = Client/Server Installation, Removal, and Configuration
8 = Exit
>1
type 1-8 <enter>;<esc> or F11=Previous Menu;F12=Help;?<enter>=Status

```

The Install/Configure screen appears.

11. Select option **3** and press **Enter**. (If you have already installed Signalware 9 and need to modify your selections, select option 5.)

```

Install/Configure
1 =[ ]Limit Installations to a Single Instance
2 = [X]Allow Multiple Installation Instances of Signalware
3 = Perform Initial Signalware Installation and Configuration
4 = Replace Signalware (replace an existing installation with new GA)
5 = Upgrade One of the Currently Installed Installation Instances (SP or ECN)
6 = Clone a Currently Installed Instance and Upgrade the Clone
>3
Type 1-6 <enter>; <esc> or F11=Previous Menu; F12=Help; ?<enter>=Status

```

The Hosts screen appears.

12. Do not enter any information. Just press and press **Enter**.

```

Hosts
The current task may be accomplished on all CEs in a cluster from a single
console. In order to do this the hostnames of the other CEs in the cluster are
needed. The other CEs must be configured to allow this host to access them
via rsh or ssh without a password. Please refer to the Installation Manual for
other required conditions. Hostnames are automatically extracted from the
current configuration. However, it appears that this is a new installation or
this CE was never configured. Please enter a space separated list of
hostnames below. Enter nothing to disable this feature.
>
Please enter the hostnames of the other CEs [ ]:

```

The Initial Install screen appears.

13. Enter **1** for Install Packages and press and press **Enter**.

```

Initial Install
1 = [ ] Install Packages
2 = [ ] Configure Platform
3 = [ ] Commission Instance
4 = [ ] Configure Nodes
5 = [ ] Start Signalware
6 = Done
>1
Type 1-6 ,enter>; <esc> or F11=Previous Menu; F12=Help; ?<enter>=Status

```

The Select Instance to Install screen appears.

14. Enter a directory to be used as the OMNI_HOME directory.

```

Select Instance to install to
No installed instances of Signalware were found in the database. Please
enter a directory value to be used as the OMNI_HOME.
Please enter a new target directory [/opt/ulcm]:

```

The packages directory screen appears.

15. Enter the full path of the directory where the packages are located. The path is expressed as `/cdrom/pathname/SignalwarePackages` or `./SignalwarePackages`.

```

Package Directory
Enter the directory path containing the packages to be installed. This may be
a local or network path or the mount point for a CDROM or DVD-ROM device. If
the packages to install are in more than one directory enter the first directory
to install. You will be prompted for additional directory paths after each set of
packages are installed.
Please enter directory containing the Signalware software
[/cdrom/cdrom0/SignalwarePackages]:

```

The Protocol Selection screen appears.

16. Select the protocol that is appropriate to your system (options 1, 2, 3, or 4). If you are upgrading from Signalware 8, select the protocol that you used with your earlier version of Signalware.

```

Protocol Selection
1 = [ ] ANSI Protocol
2 = [X] ITU Protocol
3 = [ ] CHINA Protocol
4 = [ ] JAPAN Protocol
5 = [ ] IMS (Diameter, SIP)
6 = [ ] WLAN (Authentication, SMS Gateways)
7 = [ ] CLIENT SERVER
>2
1-7 to select, 0=done, <esc> or F11/,enter>=Prev/Next Page

```

The Interface Selection screen appears.

17. Select the interfaces that are appropriate to your system (options 1, 3, or both). Only SS7 and M3UA (SIGTRAN) are supported. To select multiple interfaces, enter the corresponding numbers separated by a comma. Select 0 when done to go to the next screen.



NOTE: You must have a license for SS7, M3UA, or both.

```

Interface Selection
1 = [X] SS7 (Ulitcom Boards)
2 = [ ] M2PA
3 = [X] M3UA
4 = [ ] M2UA
5 = [ ] SUA
>1,3
1-5 to select, 0=done, <esc> or F11/,enter=>Prev/Next Page

```

If you selected SS7, the Driver Selection screen appears.

18. Select the drivers that are appropriate to your system. The PCI Bus SS7 T1/E1 - PH0301 board is shipped with SIM Server. (The PC0200 board is shipped with an earlier version of SIM Server.) If you use this board, install the OMNI-PCIM driver.

```

Driver Selection
1 = [X] OMNI-PCIM - PCI DMS Master Driver (PC02xx, CC02xx, PH03xx, MC02xx)
2 = [ ] OMNI-PCIS - PCI Bus Slave Driver (PV0100)
3 = [ ] OMNI-SBUS - SBUS Driver (SBUS V. 35)
>1
1-3 to select, 0=done, <esc> or F11/,enter=>Prev/Next Page

```

The Select Packages to Install screen appears.

19. Review the packages to be installed. Enter 0 to accept all the defaults. (Generally, you should accept the defaults and not deselect any packages.) See Table 3 on page 33 for a list of packages that will be installed for each protocol. (You will have a chance to add more packages in step 21.)



NOTE: The selected packages shown on this screen are based on the choices that you made in prior screens. For this reason, it is not advisable to deselect any packages.

The Selected Packages screen appears.

20. Enter **Y** to confirm the selection of all packages and install them.

The More Packages screen appears.

21. Enter **N**. (Y indicates that you have more packages to install.)

The Initial Install screen appears.

22. Enter **2** for Configure Platform.

The Configure Platform screen appears.

23. Type **0** to accept the install directory that you entered earlier in step 12.

```

ConfigurePlatform
1: Instance Install Directory.....: /opt/ulcm
Type Line Number to Change; Done=0 [0]
    
```

The Information Section appears.

24. Type Yes when asked Continue with configurePlatform procedure?

The CE Configuration screen appears.

25. Enter **1**. (This number must be 1.)

```

*****
*** CE Configuration ***
*****

How many CEs (1 - 4) [1]?
    
```

26. If you have an SS7 board installed, you will be led through a series of board configuration screens as shown in the following steps. Complete the board configuration screens according to the configuration of your network.

a. How many SS7 boards are on sbrha-4 (1 - 64) [1]? 1

b. Choose SS7 protocol for board 0 on sbrha-4

- 1) A7
- 2) C7
- 3) CH7
- 4) J7

Select protocol [A7]: 2

c. Choose board type for board 0 on sbrha-4

- 1) CC0200
- 2) PC0200
- 3) CC0202
- 4) CC0203
- 5) PS0204
- 6) CC0205
- 7) MC0207
- 8) PH0301

Select board type: 8 or 2

d. Link Type for sbrha-4, board 0

- 1) CHANNELIZED
- 2) UNCHANNELIZED

Enter choice [CHANNELIZED]:

- e. Port Type for sbrha-4, board 0, port 0
 - 1) E1
 - 2) T1
 Enter choice [E1]:

- f. Choose from the following items:
 - 1) double-frame
 - 2) multi-frame
 Enter choice [multi-frame]:

- g. Impedance for sbrha-4, board 0
 - 1) Balanced
 - 2) Unbalanced
 Enter choice [Balanced]:

- h. T1E1_1 Clock Source Mode for sbrha-4, board 0
 - 1) Master
 - 2) Slave
 - 3) Reference
 - 4) None
 Enter choice [Slave]:

27. The UDP/TCP Port Configuration screen prompts you for a UDP port value.

28. Enter a port number or press **Enter** to accept the default UDP port base value.

```

*****
*** UDP/TCP Port Configuration ***
*****
Signalware UDP port base value (1025 - 65535) [10000]?
  
```

The UDP/TCP Port Configuration screen prompts you for a TCP port value.

29. On the UDP/TCP Port Configuration screen, enter a port number or press **Enter** to accept the default TCP port value.

```

*****
*** UDP/TCP Port Configuration ***
*****
Choose a TCP port value. This value will be used by the Signalware
GUI and will be stored in the /etc/services file (sw_gui). This
value should be identical on all CEs. Ensure that no other services
will use this TCP port.
Signalware GUI TCP port value (1025 - 65535) [10000]?
  
```

The Configuration Section screen appears.

30. Press **Enter** to apply the configuration.

The “configurePlatform completed successfully” message appears.

31. Press **Enter**.

The Initial Install Screen appears.

32. Enter **3** to select Commission Instance (for commissioning Signalware).

The Commission Instance screen appears, showing your install directory.

33. Press **Enter** to accept the install directory.

A “commission completed successfully” message appears.

34. Press **Enter**.

The Initial Install screen appears.

35. Enter **4** to select Configure Nodes.

The Enter SHM screen appears.

36. Enter the SHM value for the user. Typically, the SHM is the same as the User ID for the UNIX user account from which you run Signalware. If you are upgrading from Signalware 8, you must use the same SHM value that you used for Signalware 8.

To find out your User ID, open a separate window as a Signalware user and type `id` at the prompt.

Example:

```
$ id
uid=900(ak) gid=1(other)
```

The returned uid of 900 is the SHM.

Enter SHM

In order to perform this operation, the system needs a SHM value.
>900
Enter SHM to Configure []:

The Signalware User screen appears.

37. Enter the username for running Signalware.

If you are upgrading from Signalware 8, you must use the same user name value that you used for Signalware 8.

Signalware User

A special user account should be used for the operation and provisioning of Signalware. Though not a rule, the user id of the account should equal the shared memory value that will be used to configure and execute Signalware. Enter the username below.
Please enter username you wish to run Signalware [deptA]:

The Configure Nodes screen appears.

38. Enter **0** to accept the values entered previously.

```

                                Configure Nodes

1: Instance Install Directory.....: /opt/ulcm
2: Shared Memory.....: 900
3: Username.....: deptA
>
Type Line Number to Change; Done=0[0]

```

The How Many Nodes screen appears.

39. Enter the number of nodes that you want to configure. (The number is usually 1.)

The Enter Node Name screen appears.

40. Enter the name of the node, such as MGW.

If you are upgrading from Signalware 8, you must use the same node names that you used for Signalware 8.

The Select Protocol screen appears.

41. Enter the protocol for the node.

If you are upgrading from Signalware 8, you must use the same node protocol for this node that you used for Signalware 8.

```

                                Select Protocol

1 = A7
2 = C7
3 = CH7
4 = J7
>1

```

The Select Protocol Type screen appears.

42. Enter the protocol type for the node. Your choice should be either 1 (for SS7), 3 for SIGTRAN, or 1,3 for both (depending on your license).

If you are upgrading from Signalware 8, you must use the same node protocol type for this node that you used for Signalware 8.

```

                                Select Protocol
Select Protocol Type

1 = SS7
2 = Broadband
3 = M3UA Application Server
4 = M3UA Sigtran Gateway
5 = M3UA SUA Application Server
6 = M3UA SUA Sigtran Gateway
7 = SUA Application Server
8 = SUA Sigtran Gateway
>3

```

43. The “Configure Nodes completed successfully” message appears.
44. When you finish configuring the node, the Initial Install screen appears.

Enter **6** to exit from the installation process without starting Signalware. (Do not start Signalware at this point.)

```

Initial Install

1 = [X]Install Packages
2 = [X]Configure Platform
3 = [X]Commission Instance
4 = [X]Configure Nodes
5 = [ ]Start Signalware
6 = Done
>6

```

The Install/Configure menu appears.

45. Press **F11**.

The Main Menu appears.

46. Enter **8** to exit.

```

Main Menu

1 = Install/Configure (Signalware is uninstalled or off-line)
2 = Online Upgrade (Signalware is installed and running)
3 = Installation Status and Reports
4 = Installation Maintenance
5 = Configuration Maintenance
6 = Start an Installed Instance of Signalware
7 = Client/Server Installation, Removal, and Configuration
8 = Exit
>8

```

The “You are about to exit...” message appears.

47. Enter **N** to exit from the installation process. (You want to exit at this point. You will start Signalware later.)

```

Main Menu

You are about to exit the Signalware Menu System. Are you sure you want to
exit? To continue using the menu system enter Y. To exit enter N.
>N
Would you like to continue the Signalware Menu System? (Y/N)[Y]:

```

Signalware Packages

Table 3 provides information about which packages are installed for each protocol that you selected in step 13 on page 26.

Table 3: Packages Installed with Each Protocol

Protocol	Packages
All protocols	OMNI OMNI-GUI OMIN-MAN OMIN-OLU OMIN-SCTP OMIN-SNMP (if using SNMP) OMNI-UTIL OMNI-GSM
ANSI	OMIN-A3 OMIN-A4 OMIN-A7X
ITU	OMIN-C3 OMIN-C4 OMIN-C7X
Chinese	OMIN-CH3 OMIN-CH4 OMIN-CH7X
Japanese	OMIN-J3 OMIN-J4 OMIN-JX
SIGTRAN	OMIN-A3M OMIN-C3M OMIN-CH3M OMIN-J3M

Removing Signalware

For information about removing (uninstalling) Signalware, see “Removing Signalware from the System” in the *Signalware Solaris Installation Manual* on the Ulticom Documentation CD that is provided with Steel-Belted Radius/SIM Server.

Starting and Stopping Signalware

Signalware needs to be configured so that it starts automatically on reboot. Once you have installed Signalware and configured it to start automatically on reboot, you can start and stop it with the start and stop commands,

Configuring Signalware to Start Automatically on Reboot

You can configure Signalware so that it restarts automatically when the system is rebooted. If you choose not to configure Signalware in this way, you need to restart Signalware with the `go.omni` command whenever the system reboots.

To configure Signalware to start automatically on reboot:

1. Change to superuser mode.

```
$su
```

2. Copy the **S91omni** file as **omni** into the directory **/etc/init.d**. The **S91omni** file is located in the **Support_Files** directory on the CD.
3. Edit the **omni** file to change the **OMNI_HOME=** line to specify the directory where Signalware is installed.

Example:

```
OMNI_HOME = /opt/omni
```

4. Edit the **omni** file to change the **SHM=** line to specify the SHM value as configured during Signalware installation.

Example:

```
SHM = 102
```

To find your user ID (which is the same as the SHM), type **id** at the prompt.

5. Change the run permissions with the following command:

```
$ chmod 755 omni
```

6. Create links into the **/etc/rc3.d** directory with the following commands.

```
$ ln -s /etc/init.d/omni /etc/rc3.d/S91omni
$ ln -s /etc/init.d/omni /etc/rc3.d/K91omni
```

7. Make sure that there is a directory called **Logs** in the directory where Signalware is installed. If not, create the **Logs** directory.

Starting and Stopping Signalware



NOTE: Ensure that you have configured Signalware to start automatically on reboot as described on page 33.

Start the Signalware system by issuing the **/etc/init.d/omni start** command.

To stop Signalware, issue the **/etc/init.d/omni stop** command.

The **\$SHM** and the **UID** should be the same on your system and you must log in using that user ID. Use the following commands to check that the **\$SHM** and the **UID** are the same. If they are not the same, contact your Juniper technical representative.

```
$echo $SHM
$id
```

Upgrading Solaris, Signalware, and SIM Server

If you have a previous version of SIM Server, you might want to upgrade to the most current version, SIM Server 5.4. This version of SIM Server is installed along with Steel-Belted Radius version 5.4.

To accomplish the upgrade, three systems need to be upgraded:

- Solaris 8 to Solaris 9
- Ulticom Signalware 8 to Ulticom Signalware 9
- SIM Server (previous version) to SIM Server 5.4

The upgrade procedure consists of the following main steps:

1. Save the existing SIM Server configuration in a safe place.
2. Save the Signalware 8 configuration in a safe place.
3. Uninstall and remove the existing version of SIM Server from the computer.
4. Uninstall and remove Signalware 8 from the computer.
5. Upgrade the operating system from Solaris 8 to Solaris 9.
6. Upgrade from Signalware 8 to Signalware 9.
7. Apply the saved Signalware configuration to Signalware 9.
8. Install Steel-Belted Radius/SIM Server 5.4.
9. Restore the SIM Server configuration.

The following sections describe each of these main steps.

Save the Existing Steel-Belted Radius/SIM Server Configuration

You need to back up the configuration of the existing SIM Server configuration so that you can restore the configuration settings after you install Steel-Belted Radius/SIM Server 5.4.

To back up the SIM Server configuration:

1. Export the database to a .rif file, as described in the “Exporting to a RADIUS Information File” section of the *Steel-Belted Radius Administration Guide*.
2. Save the .rif file to a safe place such as a networked file system or a CD.
3. Save your entire Steel-Belted Radius server directory to a safe place such as a networked file system or a CD.

Save the Signalware 8 Configuration

You need to back up the existing Signalware 8 configuration so that you can restore the configuration settings after you install Signalware 9.

1. Record your Signalware 8 license number; you will need this license number when installing Signalware 9.

To view your Signalware 8 license number, enter the following command. The system responds with your license number as shown in the example below.

```
$ cat $OMNI_HOME/bitmask
abcd002 -so3456 -e4f9p6abcde8 -9we
```

2. Convert any rc (recently changed) files to db files using the MML command **BACKUP-NODE**. To learn more about **BACKUP-NODE** enter:

```
$ man BACKUP-NODE
```

3. Use the **DFcat** command to save the contents of the following files to a safe place such as a networked file system or a CD.

```
cestart.$SHM
```

```
start.$SHM
```

```
tapdes.$SHM
```

```
db.nodename.mtp.$SHM.pri
```

```
db.nodename.sccp.$SHM.pri
```

where \$SHM represents the value of the environment variable.

Example:

```
DFcat cecstart.$SHM>SAFE_PLACE/cestart.$SHM
```

Uninstall and Remove the Existing Version of SIM Server

You need to uninstall the existing version of SIM Server so that you can install SIM Server 5.4 and then restore the SIM Server configuration.

1. Ensure that you have backed up your current Steel-Belted Radius/SIM Server configuration by following the steps in “Save the Existing Steel-Belted Radius/SIM Server Configuration” on page 35.
2. Uninstall the SNMP agent by following the instructions in the “To Uninstall the SNMP Agent” in the *Steel-Belted Radius Administration Guide*.
3. Unconfigure Steel-Belted Radius as described in the following steps:
 - a. Stop the Steel-Belted Radius daemon by issuing the following commands:

```
cd server-directory
./S90radius stop
```

- b. Uninstall the Steel-Belted Radius version 4.x software by issuing the following commands:

```
cd server-directory/install
sh install.sh -unconfig
sh install.sh -uninstall
```

4. Remove (delete) the `/radius` directory and all of its contents.

Uninstall and Remove Signalware 8

You need to uninstall and remove Signalware 8 so that you can install Signalware 9 and then restore the Signalware configuration.

To uninstall and remove Signalware 8:

1. Ensure that you have backed up the existing Signalware 8 configuration and recorded your Signalware 8 license number by following the steps in “Save the Signalware 8 Configuration” on page 36.
2. Change to superuser mode.
3. Stop Signalware (if it is running) by typing **Ctrl+C** in the Signalware (go.omni) output screen.
4. Remove the Signalware scripts by running the provided Perl script with the following command:

```
perl omni8-remove.pl
```

The Perl script is located in the following directory on the Steel-Belted Radius/SIM Server CD:

```
Support_Files\SIM_Server\Signalware8_Uninstall.
```

5. If you choose to copy the Perl script to another location before running it (and not run it directly from the CD), be sure to also copy the file called `admin` to the same place.



NOTE: The Signalware packages must be removed in the reverse order of the sequence in which they were installed. The Perl script handles removal in the correct order. The Solaris 9 operating system includes Perl in `/usr/bin/perl`.

If you prefer to remove the Signalware packages manually without the Perl script, see the “Removing Signalware” section of the *Signalware v.8.02 Service Pack 4 Installation Manual*.

Upgrade the Operating System from Solaris 8 to Solaris 9

Signalware 9 runs on the Solaris 9 operating system. For information about upgrading from Solaris 8 to Solaris 9, consult <http://www.sun.com> or your Sun representative.

Upgrading from Signalware 8 to Signalware 9

You have removed Signalware 8 and upgraded to Solaris 9. You are now ready to install Signalware 9 on a clean system.

To install Signalware 9:

1. Ensure that you have removed Signalware 8 and upgraded to Solaris 9 by following the instructions in these sections:
 - “Uninstall and Remove Signalware 8” on page 37
 - “Upgrade the Operating System from Solaris 8 to Solaris 9” on page 38
2. Ensure that the UID for your omni user is the same as the \$SHM of your saved files as described in “Save the Signalware 8 Configuration” on page 36 of this guide. For example, if you saved the file `cestart.900`, then the user’s UID must be 900. To find your user ID, type `id`.
3. Have your Signalware 8 license number ready. You recorded this number in step 1 of “Save the Signalware 8 Configuration” on page 36.
4. Follow the directions in “Installing Signalware 9.0” on page 23 of this guide.

Apply the Saved Signalware Configuration to Signalware 9

You are now ready to apply the Signalware configuration that you saved during the process for “Save the Signalware 8 Configuration” on page 36.

To restore the Signalware configuration:

1. Ensure that Signalware is not running.
2. Use the **DFconvert** command to convert the following files to Distributed Filesystem Format. DFconvert puts the files in the distributed file system. (These files were saved during the process for “Save the Signalware 8 Configuration” on page 36.)

`cestart.$SHM`

`start.$SHM`

`tapdes.$SHM`

`db.nodename.mtp.$SHM.pri`

`db.nodename.sccp.$SHM.pri`

To convert the files, use the **DFconvert** command.

Example:

```
DFconvert cestart.900.
```

3. Start Signalware by issuing the `/etc/init.d/omni start` command.

Install Steel-Belted Radius/SIM Server 5.4

You can now install Steel-Belted Radius/SIM Server.

Follow the instructions in “Installing and Running the Configuration Script” on page 17 for installing Steel-Belted Radius.

Apply the Saved SIM Server Configuration to SIM Server 5.4

You are now able to apply the Signalware configuration that you saved during the process for “Uninstall and Remove the Existing Version of SIM Server” on page 36.

1. Install the SBR Administrator application on your server.

For more information, see “Installing the SBR Administrator” in the *Steel-Belted Radius Getting Started Release 5.4* manual (provided with your software).

2. Run the `rif2xml` utility to convert the `.rif` file containing your Steel-Belted Radius database to an XML database structure.

Refer to the “RIF2XML Conversion Utility” appendix in the *Steel-Belted Radius Getting Started Release 5.4* guide for more information about running the `rif2xml` utility.

3. Restore (copy back) the following files to the `/conf` directory which is located under your radius directory. (You backed up these files during the process for “Save the Existing Steel-Belted Radius/SIM Server Configuration” on page 35.)

```
authGateway.conf
smsGateway.conf
smsulcmmg.conf
ulcmmg.conf
```

4. Merge any customized entries that you made in your old dictionary file (`.dct` file) with the new dictionary file. (You backed up the dictionary file during the process for “Save the Existing Steel-Belted Radius/SIM Server Configuration” on page 35.)
5. Restore (copy back) any files from the old (saved) `SNMP/bin/.conf` directory to the directory of the same name within the new installation. (You backed up these files during the process for “Save the Existing Steel-Belted Radius/SIM Server Configuration” on page 35.)
6. Merge any settings that you customized in your old Steel-Belted Radius configuration files (`*.ini`, `*.aut`, `*.dir`, `*.pro`, `*.rr`, `*.gen`, `*.acc`) to the new configuration files in the directory in which Steel-Belted Radius is installed.

To “merge” settings, compare the old configuration files with the new configuration files of the same name. Copy any settings that you customized in the old files into the new files. (Do not simply use the entire old file because you need to preserve the old settings while retaining the settings in the new files.)

7. Copy any customized MIB files to the new `radius/snmp/mibs` directory.
8. Run the SBR Administrator.

Refer to the *Steel-Belted Radius Administration Guide 5.4* for information about how to use the SBR Administrator.

9. Select **File > Import** to import the converted XML database generated by the `rif2xml` utility into Steel-Belted Radius.

Refer to the *Steel-Belted Radius Administration Guide Release 5.4* for information about how to import information into Steel-Belted Radius.

10. Use the SBR Administrator to apply the EAP settings that you were using before the upgrade.
11. Restart Steel-Belted Radius.

Chapter 3

Configuring SS7/IP Networks

Signalware provides the ability to communicate across SS7 networks. SIGTRAN (provided as a module of Signalware) allows SS7 communication over IP networks.

When configuring the communication pathways between Steel-Belted Radius and the network servers, you configure certain files based on your choices of the following:

- Type of network equipment with which you communicate. (See Table 4 for information of which type of equipment is used.)
 - HLR
 - MSC
- Type of network.
 - SS7
 - SS7 over IP (SIGTRAN).

Table 4: Network Equipment Used for Authorization

Action Needed to Process Access-Request	Network Equipment
Obtain SIM triplets	HLR
Obtain AKA quintets	HLR
Send SMS text message containing password	MSC
Obtain IMSI (given the MSISDN)	HLR
Obtain MSISDN (given the IMSI)	HLR
Obtain Authorization string	HLR or database

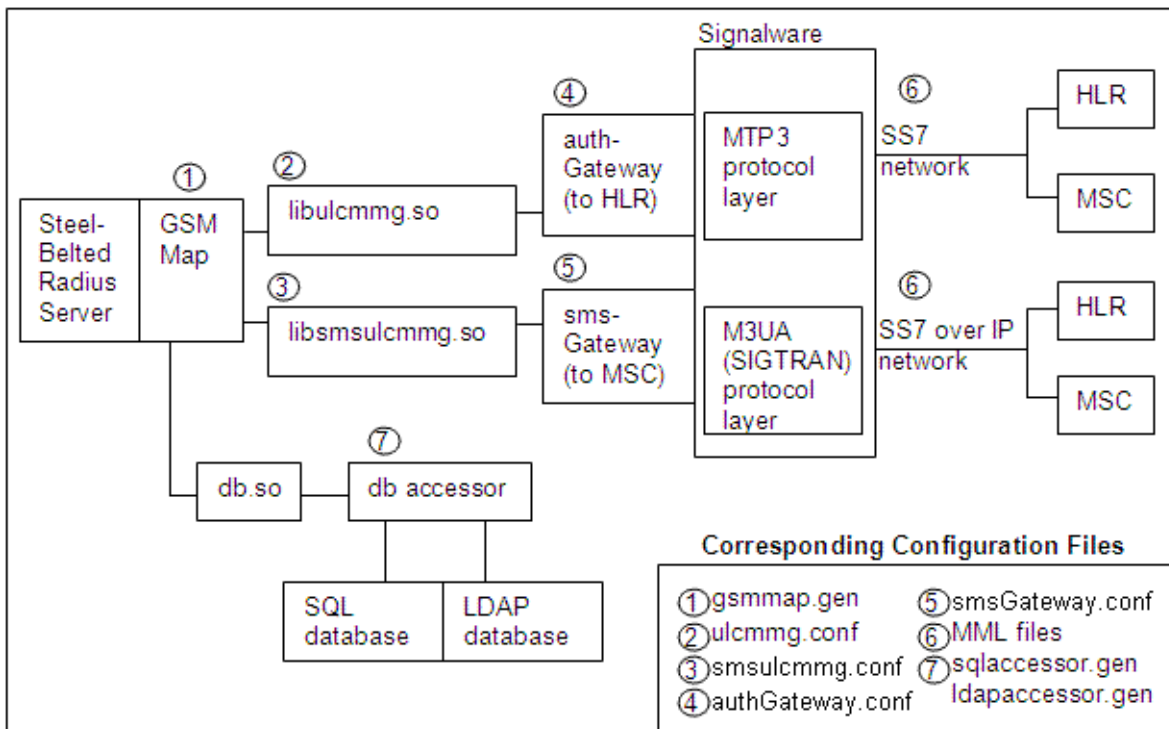


NOTE: If quintets are received but triplets are needed, then SIM Server converts the quintets to triplets according to specification 3G TS 33.102, available at <http://www.3gpp.org>.

Communication Pathways and Corresponding Files

Figure 7 shows the communication pathways between the Steel-Belted Radius server and the network equipment. Figure 7 also lists the files that need to be configured for each segment in the communication pathways.

Figure 7: SIM Server Communication Modules/Libraries and Corresponding Configuration Files



Signalware MML Commands

Once Signalware is installed, you can configure and provision Signalware using commands sent to the Signalware system. (See “Installing Signalware 9.0” on page 23.) These commands are in a format called MML (Man-Machine Language).

You can input MML commands individually using the SWMML program, or save them in a file. The procedures shown in this chapter assume that you save the MML commands to .mml text files and execute them as described in “Loading the MML Configuration Settings” on page 58.

The basic activities that require MML commands are:

- Setting up link sets, links, and routes
- Configuring the authGateway or smsGateway location and startup information
- Loading the MML configuration settings

All the MML commands are described in the *Ulticom Signalware Operator's Reference Manual*, provided with SIM Server. You can also view a list of all the MML commands and definitions by typing:

```
$ man MML_Intro
```

You can view specific information about any MML command by typing `man cmdname`. For example:

```
$ man CRTE-LSET
```

Configuration Activities

To configure the communication pathways, perform the following activities. Each of these activities is described in its own individual section.

- Install Signalware. See “Installing Signalware 9.0” on page 23.
- Start Signalware. See “Starting and Stopping Signalware” on page 33.
- Define links, link sets, and route sets with MML commands for SS7 or SIGTRAN. See “Defining Links, Link Sets, and Route Sets.”
- Configure the authGateway application for HLR communication. See “Configuring the authGateway Application for HLR Communication” on page 46.
- Configure the SMSGateway application for MSC communication. See “Configuring the SMSGateway Application for MSC Communication” on page 52.
- Load the MML configuration settings. See “Loading the MML Configuration Settings” on page 58.

Defining Links, Link Sets, and Route Sets

Links identify point-to-point connections of an adjacent node. Link sets are sets of parallel links that can be used for load sharing. Routes identify the final node destination. (The order in which they are defined is important.) Figure 8 on page 44 provides a simple example of links, link sets, and route sets. Note that in the case of a SIGTRAN installation, the link set will most likely consist of a single link because redundancy might not be needed.

You use MML commands to set up links, link sets, and routes to identify the path between Steel-Belted Radius, the adjacent link, and the final destination (HLR or MSC).

Setting up links, link sets, and route sets involves the MML commands shown in Table 5. For more information about the syntax and usage of the MML commands, see “Signalware MML Commands” on page 42.

Table 5: MML Commands for Defining Links, Link Sets, and Route Sets

Action	MML Command for SS7 Networks	MML Command for SIGTRAN Networks
Identify the local point code and network indicator (NI)	CRTE-OSPC	CRTE-OSPC
Create a link set and assign it a point code	CRTE-LSET	CREATE-M3UA-LSET
Create one or more links that belong to the link set	CRTE-SLK	CREATE-M3UA-SLK
Create a route set that identifies the final destination	CRTE-RSET	CRTE-M3UA-RKEY
Allow the route set to be used	ALW-RSET	ALW-RSET
Activate the links	ACTV-SLK	ACTV-SLK

Example MML Commands

The following examples illustrate the MML commands used to create links, link sets, and route sets.

Figure 8: Links, Link Sets, and Route Sets

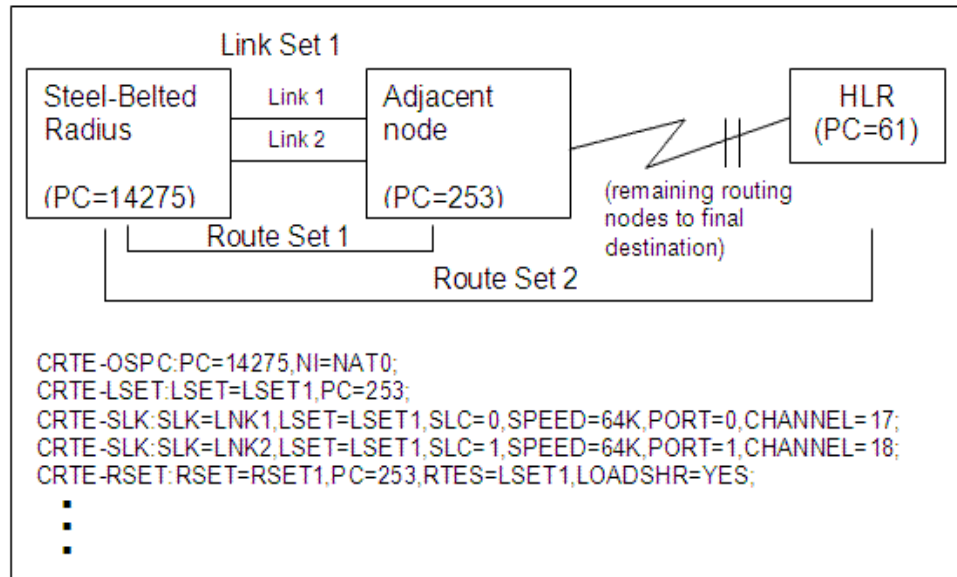
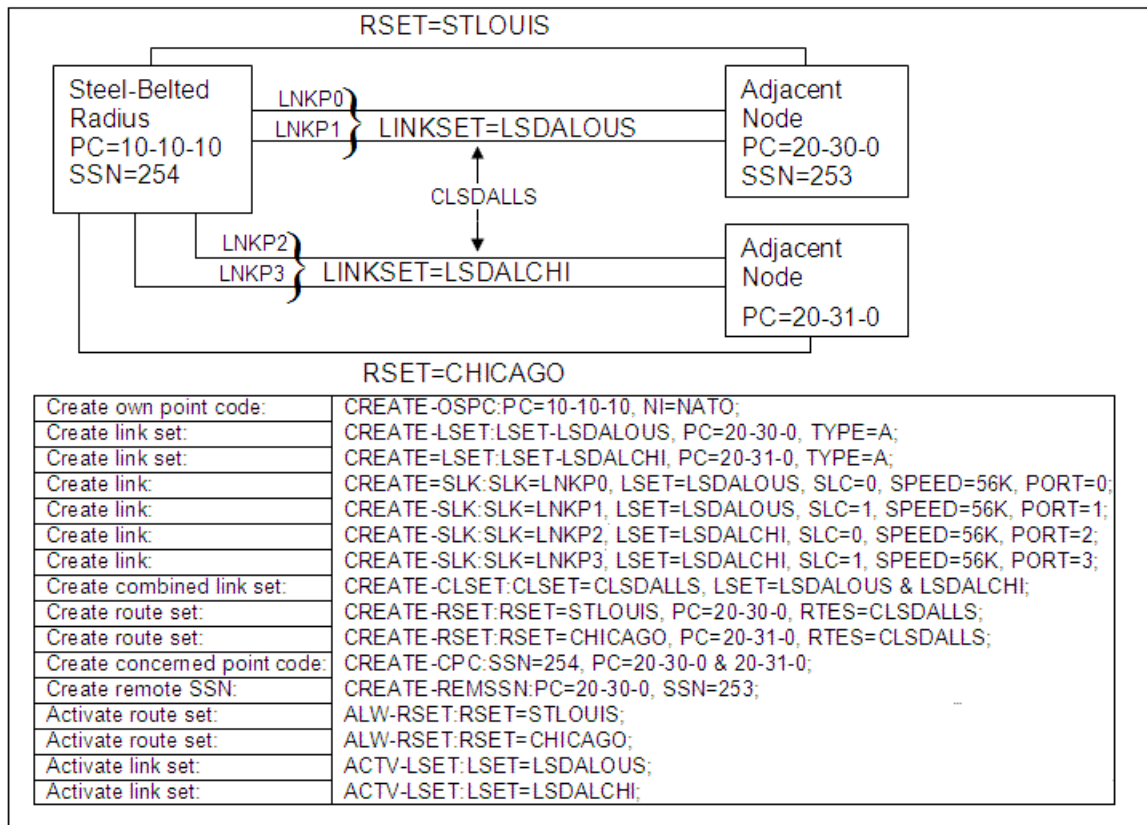


Figure 9: MML Provisioning

**SS7 Example: Creating Links, Link Sets, and Route Sets**

```

CRTE-OSPC:PC=14275,NI=NATO;
CRTE-LSET:LSET=LSET1,PC=253;
CRTE-SLK:SLK=LNK1,LSET=LSET1,SLC=0,SPEED=64K,PORT=0,CHANNEL=17
CRTE-RSET:RSET=RSET1,PC=253,RTES=LSET1,LOADSHR=YES;
CRTE-RSET:RSET=RSET2,PC=61,RTES=LSET1,LOADSHR=YES;
ALW-RSET:RSET=RSET1;
ALW-RSET:RSET=RSET2;
ACTV-SLK:SLK=LNK1;

```

SIGTRAN Example: Creating Links, Link Sets, and Route Sets

```

CREATE-OSPC:PC=4004,NI=INTO;
CREATE-M3UA-LSET:LSET=IPLSET,TYPE=IPSP-IPSP, RADDR=207.46.20.60,PC=5005;
CREATE-M3UA-SLK:SLK=IPSLK,LSET=IPLSET,LADDR=72.5.124.61,RADDR=207.46.20
.60,MODE=CONNECT,LPORT=2906;
ACTIVATE-M3UA-SLK:SLK=IPSLK;

CREATE-RSET:RSET=IPRSET,PC=5005,RTES=IPLSET;
ALLOW-RSET:RSET=IPRSET;

CREATE-M3UA-RKEY:RKEY=RK1,TYPE=STATIC-AS,TRAFFIC-MODE=LOADSHARE,LSET=I
PLSET,DPC=4004,SI=SCCP,SSN=1&251&252;
ACTIVATE-M3UA-RKEY:RKEY=RK1;

```

Configuring the authGateway Application for HLR Communication

The authGateway application manages all communication between Steel-Belted Radius and the HLR. The authGateway application also implements the MAP (Mobile Application Port) protocol and MAP messages that get sent through the Signalware protocol stack and out to the HLR and back.

Configuration of the authGateway application requires the activities described in the following sections of this chapter:

- “Configuring the authGateway Routing Location Information” on page 46
- “Configuring the ulcmmg.conf File” on page 52
- “Configuring the authGateway Startup Information” on page 50
- “Configuring the ulcmmg.conf File” on page 52

Configuring the authGateway Routing Location Information

This activity assigns the local routing options and the remote routing options using the MML commands shown in Table 6.

Table 6: MML Commands for Configuring the authGateway Routing Location Information

MML Command	Description
CREATE-CPC	Identify the concerned point code (CPC), which is the destination point code and the local application (authGateway).
CREATE-REMSSN	Identify the point code of the HLR and the remote application.
CREATE-GT	Create a global title translation for the remote HLR (if Global Title routing is used).

For more information about the syntax and usage of the MML commands, see “Signalware MML Commands” on page 42.

Example 1 — Global Title Routing Using Global Title Identification

In the following example, these actions take place:

Line 1 (CREATE-GT): Global Title type translation will be used so that digits 22201 will be sent to PC,SSN = 61,6. RI = GT tells 61,6 that it needs to find the next routing hop for the request.

```
CREATE-GT:TT=0,NP=ISDN-MOB,NA=INT,DIG="22201",PC=61,SSN=6,RI=GT;
```

Example 2 — PCSSN Routing Using Point Code Identification

In the following example, these actions take place:

Line 1 (CREATE-CPC): authGateway is assigned a subsystem number (SSN) of 7 on the local host and the concerned point code on the HLR is identified as 61.

Line 2 (CREATE-REMSSN): The subsystem number (application) on the remote host is identified as 6.

Line 3 (CREATE-GT): Global Title type translation will be used so that digits 22201 will be sent to PC,SSN = 61,6. RI = PCSSN indicates that digits 22201 are handled by PC,SSN = 61,6.

```
CREATE-CPC:PC=61,SSN=7;
CREATE-REMSSN:PC=61,SSN=6;
CREATE-GT:TT=0,NP=ISDN-MOB,NA=INT,DIG="22201",PC=61,SSN=6,RI=PCSSN;
```



NOTE: MML commands are saved in MML files which can be loaded into Signalware. See “Loading the MML Configuration Settings” on page 58.

Configuring the *authGateway.conf* File

The *authGateway.conf* file is the configuration file that specifies remote routing and authorization options of the *authGateway* application.

- Remote routing options control how the remote HLR is addressed based on the incoming IMSI.
- Authorization options control whether or not a subscriber requesting an account is authorized for WLAN access, and which Steel-Belted Radius profile or native user is used.

Remote Routing Options

Each line in the *authGateway.conf* file represents a target HLR, where each HLR has its own routing options and authorization options. Indicate each HLR listed in this file with the initial digits of the subscriber password, specified by the *odigits* option.

Table 7 lists the remote routing options for the *authGateway.conf* file.

Table 7: *authGateway.conf* Remote Routing Options

Option	Purpose
bs	Bearer Service. See “Authorization Options” on page 48.
msisdn	The <i>msisdn</i> option can be used in place of <i>ndigits</i> and <i>odigits</i> when no translation is required. See Example 2 on page 50.
ndigits	Replacement digits for numbering plan translation (hybrid IMSI).
odb	Operator-Determined Barring. See “Authorization Options” on page 48.

Table 7: authGateway.conf Remote Routing Options (continued)

Option	Purpose
odigits	<p>Initial digits of IMSI or password for this HLR. For each request, the first digits of the IMSI are compared with odigits. The first line of the configuration file that matches is selected for the current request.</p> <p>If the routing indicator (rri) is 0 (Global Title), the leading digits are replaced with the new digits (ndigits) to perform the numbering plan translation.</p> <p>Example of direct replacement:</p> <p>If the rule is “odigits 12345 ndigits 98765” and the IMSI is 123456789012345, the resulting digits will be 987656789012345.</p> <p>Example of wildcard replacement:</p> <p>If the rule is “odigits 12345* ndigits 98765” and the IMSI is 123456789012345, the resulting digits will be 98765.</p>
rgti	(Global Title only) GTI value. 4 for C7; 2 for A7. (Usually 4.)
rnai	(Global Title only) Nature of Address Indicator.
rnp	<p>(Global Title only) Numbering Plan.</p> <p>Acceptable values are:</p> <ul style="list-style-type: none"> 1 — ISDN/Telephony 3 — DATA 4 — TELEX 5 — Maritime Mobile 6 — Land/Mobile 7 — ISDN/Mobile 10 — British Telecom special 1 11 — British Telecom special 2 14 — Private Network
rpc	Remote Point Code. Point Code of HLR or MSC.
rri	Routing indicator - 0 for GT (Global Title), 1 for PC/SSN (Point Code/Subsystem Number).
rssn	Subsystem Number of HLR.
rtt	(Global Title only) Translation Type (usually 0).
ts	Teleservice. See “Authorization Options” on page 48.

Authorization Options

The HLR database includes authorization information that is assigned to each subscriber. Three authorization designations are relevant to Steel-Belted Radius/SIM Server:

- BS (Bearer Service)
- TS (Teleservice)
- ODB (Operator-Determined Barring)

You can specify subscriber HLR authorization (and barred service) designations in the MAP Gateway `authGateway.conf` file.



NOTE: It is possible to disable authorization completely from EAP-SIM (not fetch subscriber profile information from the HLR and not perform a SQL/LDAP query). For instructions about disabling authorization, see “Disabling Authorization from EAP-SIM” on page 63.

Each line in the `authGateway.conf` file corresponds to an HLR in your network. Each line also specifies all potential authorization (and barred service) settings for any subscribers on this HLR.

Steel-Belted Radius/SIM Server uses the service authorization information that you list for each HLR in `authGateway.conf`, as follows:

- When a TS or BS designation is assigned to a subscriber entry in the HLR database, Steel-Belted Radius/SIM Server permits the subscriber the designated class of WLAN service upon authorization request.
- When an ODB designation is assigned to a subscriber, Steel-Belted Radius/SIM Server denies the subscriber WLAN service upon authorization request.
- When you do not specify service designations for a HLR listed in `authGateway.conf`, then all subscribers on that HLR are authorized for WLAN service.
- You can specify up to six authorization strings of each type (TS, BS, or ODB) on any given line of `authGateway.conf`.

You can specify the service designations in `authGateway.conf`, as follows:

```
bs n1:auth1
ts n2:auth2
odb n3:auth3
```

Here, *ni* (*i=1,2,3*) is a decimal integer that specifies the setting, and *authi* (*i=1,2,3*) is the string returned from the MAP Gateway to Steel-Belted Radius/SIM Server.

For example, you might specify the potential subscriber designations on one HLR with the following text in `authGateway.conf`:

```
bs 26:B1A ts 33:TS21 odb 128:bar
```



NOTE: If you require any HLR authorization strings to define different classes of service for your subscribers, you must also specify those TS, BS, and ODB authorization strings in some Steel-Belted Radius/SIM Server configuration files. See “`smsprov.aut` [ProfileMap] Section” on page 131 and “`simauth.aut` [ProfileMap] Section” on page 127 for information about how to match these strings to Steel-Belted Radius/SIM Server variables.

Example 1: authGateway.conf file (Note: Lines are wrapped.)

```
odigits 2310 ndigits 2324 rnai 4 rnp 7 rgti 4 rtt 0 rri 0 rpc 3003 rsn 251 bs 12:gold
bs 23:silver ts 91:bronze ts 92:red ts 93:green odb 1:black aqua
```

```
odigits 31026 ndigits 32476 rnai 4 rnp 7 rgti 4 rtt 0 rri 1 rpc 3003 rsn 253 bs
23:morning bs 24:afternoon ts 1:night
```

Example 2 authGateway.conf file (Note: Lines are wrapped.)

In this global title example, odigits and ndigits are the same and do not require translation. You can use the msisdn option in place of ndigits and odigits when no translation is required.

```
msisdn 31026 rnai 4 rnp 7 rgti 4 rtt 0 rri 0 rpc 3003 rsn 251 bs 12:gold bs 23:silver
ts 91:bronze ts 92:red ts 93:green odb 1:black aqua
```

Configuring the authGateway Startup Information

The CREATE-PROCESS and START-PROCESS MML commands start the authGateway (by calling authGateway.conf), using options that you specify.

Table 8 describes the MML commands needed to configure the start of authGateway.

Table 8: MML commands for configuring the start of authGateway

MML Command	Description
CREATE-PROCESS	Identify the authGateway configuration file and the authGateway options.
START-PROCESS	Start the process.

For more information about the syntax and usage of the MML commands, see “Signalware MML Commands” on page 42. See “Loading the MML Configuration Settings” on page 58 for information about executing the MML commands.

Table 9 lists the options that you can use with the CREATE-PROCESS command.

Table 9: authGateway Process Options Used with CREATE-PROCESS

Option	Description
appctx	MAP protocol revision (2 or 3).
conf	Path and name of the authGateway configuration file. The default file is OMNI_HOME/conf/authGateway.conf.
debug	Sets a debug level. Use the following: -debug 0xff
host	Local hostname.
invkretry	Number of invoke retry.
invktimeout	Duration of invoke timeout in seconds.
lgti	(Global Title only) Local GTI value, usually 4 for C7 and 2 for A7.

Table 9: authGateway Process Options Used with CREATE-PROCESS (continued)

Option	Description
lmsisdn	(Global Title only) MSISDN of this local node.
lnai	(GT only) Nature of Address Indicator. Indicates the scope of the address value, such as whether it is an international number (includes country code) or a national number (no country code). 1 Subscriber Number — no area code (example: 5551234) 2 unused 3 National Significant Number — no country code (example: 2015551234) 4 International Number — includes country code (example: 12015551234)
lnp	(Global Title only) Local Numbering Plan. Acceptable values are: 1 — ISDN/Telephony 3 — DATA 4 — TELEX 5 — Maritime Mobile 6 — Land/Mobile 7 — ISDN/Mobile 10 — British Telecom special 1 11 — British Telecom special 2 14 — Private Network
lpc	Local Point Code (PC).
lri	Routing indicator - 0 for GT (Global Title), 1 for PC/SSN.
lssn	Local Subsystem Number (SSN) (required).
ltn	(Global Title only) Local Translation Type. Generally in a live network TT will always be 0.
max_requests	The maximum number of simultaneous MAP dialogs.
monitor	Activates Message Activity Monitor.
name	Name of the process.
no_rst	Disables automatic restart of process.
node	Node name.
port	Port number used by the SCTP association with the client.
prot	Variant used (C7, A7, or CH7).
trace	Enables debug tracing and displays the trace information on the console. (Consists of a trace of all MAP messages that are formatted and sent down the stack.) Use the tracefile option to capture the trace information to a file.
tracefile	Captures the trace information to a file. The filename follows the <code>-tracefile</code> switch. Include the directory in the filename.

Example—Creating and Starting the authGateway Process

Notice that the SSN = 7 in the CREATE-CPC of the previous example (see Example 1 in “Configuring the authGateway Routing Location Information” on page 46) becomes the lssn (local subsystem number) in the CREATE-PROCESS command of this example. The SSN = 6 in the CREATE-REMSSN command of the previous example (see Example 1 in “Configuring the authGateway Routing Location Information” on page 46) becomes the -rssn (remote subsystem number) in this example.

It is recommended that the EXEC command use an absolute (full) pathname.

(Note: Lines are wrapped.)

```
CREATE-PROCESS:NAME="GMT", CE="sbrss7",
EXEC="/opt/funk/radius/authGateway-name GMT -port 2000 -host sbrss7
-node MGW -prot C7 -conf /opt/funk/radius/conf/authGateway.conf -lri 0 -lpc
14275 -lssn 7 -rssn 6 -lmsdn 393558817298 -lgti 4 -lnp 7 -litt 0 -lnai 4 -trace";
START-PROCESS:NAME="GMT", CE="sbrss7";
```



NOTE: MML commands are saved in MML files which can be loaded into Signalware. See “Loading the MML Configuration Settings” on page 58.

Configuring the ulcmmg.conf File

The ulcmmg.conf file establishes the connection between the authGateway application and Steel-Belted Radius.

The ulcmmg.conf file consists of two lines, as shown in the following example. Modify the ulcmmg.conf file shipped with SIM Server so that the LOCAL_HOST (Steel-Belted Radius) and REMOTE_HOST (Signalware system) values identify their DNS names and TCP port numbers. If you specify a DNS name for a local or remote host, you can enter the host’s IP address in brackets as a backup.

Example

```
LOCAL_HOST myhost.com:2001
REMOTE_HOST signalwarehost.com:2000 [172.25.97.230]
```

Configuring the SMSGateway Application for MSC Communication

The SMSGateway application manages all communication between Steel-Belted Radius and the MSC. The SMSGateway application also implements the MAP (Mobile Application Port) protocol and MAP messages that get sent through the Signalware protocol stack and out to the MSC (and back).

Configuration of the SMSGateway application requires the following activities:

- Configuring the SMSGateway Routing Location Information
- Configuring the SMSGateway.conf File
- Configuring the SMSGateway Startup Information

- Configuring the smsulcmmg.conf File

Each of these activities is described in the following sections.

Configuring the SMSGateway Routing Location Information

This activity assigns the local routing options and identifies the point code of the MSC and the remote application using the MML commands shown in Table 10.

Table 10: MML Commands for Configuring the SMSGateway Routing Location Information

MML Command	Description
CREATE-CPC	Identify the concerned point code (CPC), which is the destination point code and the local application (authGateway).
CREATE-REMSSN	Identify the point code of the HLR and the remote application.
CREATE-GT	Create a global title translation for the remote HLR (if Global Title routing is used).

For more information about the syntax and usage of the MML commands, see “Signalware MML Commands” on page 42.

Example

Line 1 (CREATE-CPC): SMSGateway is assigned a subsystem number (SSN) of 242 on the local host and the concerned point code on the MSC is identified as 3003.

Line 2 (CREATE-REMSSN): The subsystem number (application) on the remote host is identified as 241.

Line 3 (CREATE-GT): Global Title type translation will be used so that digits 987 will be sent to PC,SSN = 3003,241. RI is not supplied in the CREATE-GT command and therefore defaults to DEF (do not modify) and uses the routing setting of the incoming message.

```
CREATE-CPC:PC=3003,SSN=242;
CREATE-REMSSN:PC=3003,SSN=241;
CREATE-GT:TT=0, NP=ISDN-TEL, NA=INT, DIG="987", PC=3003, SSN=241;
```



NOTE: MML commands are saved in MML files which can be loaded into Signalware. See “Loading the MML Configuration Settings” on page 58.

Configuring the SMSGateway.conf File

The SMSGateway.conf file is the configuration file that specifies routing and options of the SMSGateway application.

Table 11: Local Routing Options for SMSGateway.conf

Option	Description
lpc	Local Point Code (PC) (required).
lssn	Local Subsystem Number (SSN) (required).

Table 11: Local Routing Options for SMSGateway.conf (continued)

Option	Description
oatype	Originating Address type: 0 — Unknown type 1 — International number 2 — National number 3 — Network-specific number 4 — Subscriber number 5 — Alphanumeric number 6 — Abbreviated number
oanp	Originating Address numbering plan: 0 — Unknown 1 — ISDN/telephone numbering plan (E.164/E.163) 3 — Data numbering plan (X.121) 4 — Telex numbering plan 8 — National numbering plan 9 — Private numbering plan 10 — ERMES numbering plan (ETSI DE/PS 3 01-3) Refer to 3G TS 23.040 for more information on numbering plan values.
oadigits	Originating Address digits.

Table 12: Remote Routing Options for SMSGateway.conf Process

Option	Description
rgti	Remote GTI value, usually 4 for C7 and 2 for A7.
rnai	Nature of Address Indicator. Indicates the scope of the address value, such as whether it is an international number (includes country code) or a national number (no country code). 1 — Subscriber Number with no area code (example: 5551234) 2 — unused 3 — National Significant Number with no country code (example: 2015551234) 4 — International Number including country code (example: 12015551234)
rnp	Remote Numbering Plan. Acceptable values are: 1 — ISDN/Telephony 3 — DATA 4 — TELEX 5 — Maritime Mobile 6 — Land/Mobile 7 — ISDN/Mobile 10 — British Telecom special 1 11 — British Telecom special 2 14 — Private Network

Table 12: Remote Routing Options for SMSGateway.conf Process (continued)

Option	Description
rpc	Remote Point Code. Point Code of MSC.
rri	Remote Routing indicator - 0 for GT (Global Title), 1 for PC/SSN.
rssn	Subsystem Number of MSC.
rtt	Remote Translation Type. Generally in a live network TT will always be 0.

Example

```
DestAddress rnai 4 rnp 1 rgti 4 rtt 0 rri 0
LocalAddress lpc 3003 lssn 7 ldigits 1234 oatype 1 oanp 1 oadigits 7654321
```

Configuring the SMSGateway Startup Information

The CREATE-PROCESS and START-PROCESS MML commands start the smsGateway (by calling smsGateway.conf), using options that you specify.

Table 13 describes the MML commands needed to configure the start of smsGateway.

Table 13: MML Commands for Configuring the Start of SMSGateway

MML Command	Description
CREATE-PROCESS	Identify the SMSGateway configuration file and the SMSGateway options.
START-PROCESS	Start the process.

For more information about the syntax and usage of the MML commands, see “Signalware MML Commands” on page 42

Table 14 lists the options that you can use with the CREATE-PROCESS command for local address options. Table 15 on page 57 lists the options that you can use with the CREATE-PROCESS command for remote address options.

Table 14: Routing Options for the SMSGateway Process Used with CREATE-PROCESS Command

Option	Description
appctx	MAP protocol revision (2 or 3).
conf	Path and name of the SMSGateway configuration file. The default file is OMNI_HOME/conf/SMSGateway.conf.
debug	Sets a debug level. Use the following: -debug 0xff
host	Local hostname.
invkretry	Number of invoke retry.
invktimeout	Duration of invoke timeout in seconds.
ldigits	MSISDN of the local node.
lgti	Local GTI value, usually 4 for C7 and 2 for A7.
lmsisdn	MSISDN of the local node. (Same as ldigits.)

Table 14: Routing Options for the SMSGateway Process Used with CREATE-PROCESS Command (continued)

Option	Description
lnai	Local Nature of Address Indicator. Indicates the scope of the address value, such as whether it is an international number (includes country code) or a national number (no country code). 1 — Subscriber Number with no area code (example: 5551234) 2 — unused 3 — National Significant Number with no country code (example: 2015551234) 4 — International Number including country code (example: 12015551234)
lnp	Local Remote Numbering Plan. Acceptable values are: 1 — ISDN/Telephony 3 — DATA 4 — TELEX 5 — Maritime Mobile 6 — Land/Mobile 7 — ISDN/Mobile 10 — British Telecom special 1 11 — British Telecom special 2 14 — Private Network
lnp	Local Numbering Plan Acceptable values are: 1 — ISDN/Telephony 3 — DATA 4 — TELEX 5 — Maritime Mobile 6 — Land/Mobile 7 — ISDN/Mobile 10 — British Telecom special 1 11 — British Telecom special 2 14 — Private Network
lpc	Local Point Code (PC) (required).
lri	Routing indicator - 0 for GT (Global Title), 1 for PC/SSN.
lssn	Local Subsystem Number (SSN) (required).
ltn	Local Translation Type. Generally in a live network TT will always be 0.
max_requests	Maximum number of simultaneous MAP dialogs.
max_RoutingInfo	Maximum number of routing information cached.
name	Name of the process.
node	Node name.
port	Port number used by the SCTP association with the client.
prot	Variant used (C7, A7 or CH7).
routingInfoExp	Validity time for routing information.

Table 14: Routing Options for the SMSGateway Process Used with CREATE-PROCESS Command (continued)

Option	Description
smsc	Service center address.
tp_oa	<p>Originating address information in SMS header.</p> <p>The tp_oa field is made up of three fields: oatype, oanp, and oadigits. It is recommended to use these three fields as local routing options for SMSGateway.conf, rather than the tp_oa field.</p> <p>For information about oatype, oanp, and oadigits, see Table 11 on page 53.</p>
trace	<p>Enables debug tracing and displays the trace information on the console. (Consists of a trace of all MAP messages that are formatted and sent down the stack.)</p> <p>Use the tracefile option to capture the trace information to a file.</p>
tracefile	Captures the trace information to a file. The filename follows the -tracefile switch. Include the directory in the filename.

Table 15: Remote Routing Options for the SMSGateway Process used with CREATE-PROCESS Command

Option	Description
rmai	<p>Remote Nature of Address Indicator. Indicates the scope of the address value, such as whether it is an international number (includes country code) or a national number (no country code).</p> <p>1 — Subscriber Number with no area code (example: 5551234)</p> <p>2 — unused</p> <p>3 — National Significant Number with no country code (example: 2015551234)</p> <p>4 — International Number including country code (example: 12015551234)</p>
rnp	<p>Remote Numbering Plan</p> <p>Acceptable values are:</p> <p>1 — ISDN/Telephony</p> <p>3 — DATA</p> <p>4 — TELEX</p> <p>5 — Maritime Mobile</p> <p>6 — Land/Mobile</p> <p>7 — ISDN/Mobile</p> <p>10 — British Telecom special 1</p> <p>11 — British Telecom special 2</p> <p>14 — Private Network</p>
rgti	Remote GTI value, usually 4 for C7 and 2 for A7.
rtt	Remote Translation Type. Generally in a live network TT will always be 0.

Example—Creating and Starting the smsGateway Process

It is recommended that the EXEC command use an absolute (full) pathname.

(Note: Lines are wrapped.)

```
CREATE-PROCESS:NAME="SMS", CE="quark",
EXEC="/opt/funk/radius/smsGateway -name SMS -port 2004 -host quark -node MGW
-prot C7 -conf /opt/funk/radius/conf/smsGateway.conf -lri 1 -lpc 2730 -lssn 242 -rssh
241 -appctx 3";
START-PROCESS:NAME="SMS", CE="quark";
```

Configuring the smsulcmmg.conf File

The `smsulcmmg.conf` file establishes the connection between the SMSGateway application and Steel-Belted Radius.

The `smsulcmmg.conf` file consists of two lines, as shown in the example below. Modify the `smsulcmmg.conf` file shipped with SIM Server so that the `LOCAL_HOST` (Steel-Belted Radius) and `REMOTE_HOST` (Signalware system) values identify their DNS names and TCP port numbers. If you specify a DNS name for a local or remote host, you can enter the host's IP address in brackets as a backup.

Example

```
LOCAL_HOST quark:2005
REMOTE_HOST quark:2004 [172.25.97.230]
```

Loading the MML Configuration Settings

The files containing MML commands need to be loaded into Signalware. Enter an SWMML command for each `.mml` file that you created.

In the following example, three files are loaded into Signalware. The file called `links.mml` sets up the links, link sets, and routes. The file called `config_authgateway.mml` configures the `authGateway` application. The file called `start_authgateway.mml` configures the `authGateway` startup information.

Note that these file do not exist and are used here as an example, You need to create your own files for accomplishing the tasks of creating links, configuring the `authGateway` (or `smsGateway`) application, and starting the `authGateway` (or `smsGateway`) application.

Example

```
$ swmml -node MGW -f links.mml
$ swmml -node MGW -f config_authgateway.mml
$ swmml -node MGW -f start_authgateway.mml
```



NOTE: Signalware remembers the MML configuration commands and uses the same configuration each time you start Signalware. To reconfigure the nodes, use the `configureNodes` command with the `-f` or `-clean` or `-realclean` options.

Chapter 4

Configuring the Steel-Belted Radius Files

This chapter describes the configuration of the Steel-Belted Radius files that are involved in SIM Server Access-Request authentication. These files include:

- `gsmmap.gen`
- `sqlaccessor.gen`
- `sqlaccessorjdbc.gen`
- `ldapaccessor.gen`

Configuring the `gsmmap.gen` File

The `gsmmap.gen` file enables you to configure authentication settings by realm. This file consists of several sections that you need to configure, including:

- [Bootstrap] section
- [Settings] section
- [Realms] section
- Each realm section
- Target module sections

This chapter describes each of these configuration sections.

[Bootstrap] Section

The [Bootstrap] section of the `gsmmap.gen` file enables the `gsmmap.gen` file to function. Table 16 shows the fields of the [Bootstrap] section.

Table 16: `gsmmap.gen` [Bootstrap] Fields

Field	Description
LibraryName	Specifies the name of the library called when <code>gsmmap</code> runs. Default value is <code>gsmmap</code> .

Table 16: gsmmap.gen [Bootstrap] Fields (continued)

Field	Description
Enable	Set to 1 to enable the features described in this file. Set to 0 to disable the features described in this file. Default value is 1.
DependsOn	Specifies the names of the module or modules (comma separated) that must be running for the MAP module to work. Default values is: ldapaccessor.gen, sqlaccessor.gen

Example

```
[Bootstrap]
LibraryName=gsmmap
Enable=1
DependsOn=ldapaccessor.gen,sqlaccessor.gen
```

[Settings] Section

The [Settings] section controls how log information is handled.

Table 17: gsmmap.gen [Settings] Fields

Field	Description
ConfigLog	Method for capturing log information. ConfigLog = None means that configuration information will not be captured. ConfigLog = ConsoleAndLog sends the log information to both the console and the log. ConfigLog = Console sends the log information to the log file only. ConfigLog = Log sends the log information to the log file only. Default is ConsoleandLog.

Example

```
[Settings]
ConfigLog=Log
```

[Realms] Section

The [Realms] section of the `gsmmap.gen` file contains a list of realms for which you specify authentication instructions. When an Access-Request is received, Steel-Belted Radius handles the request in different ways, depending on the settings in the [Realms] section. For example, requests from the ABC.com realm might require the IMSI retrieved from the LDAP database for authentication, Requests from the XYZ.com realm might require the AKA from the MAP Gateway for authentication.

You can specify realms in several ways:

- By name – You can specify realms directly by listing names of authorized realms. Example: `abc.com`.
- By alias – You can create an alias for a realm by specifying the realm alias and realm name. Example: `realm1=abc.com`
- By wild-carded alias – You can create an alias that includes a wildcard to permit authentication for multiple realms. Example: `realm2=*abc.com` or `realm=abc.*`
- By unmatched realm – You can create an alias that applies to all realms that do not match any specified realm. Example: `CatchAllRealm=*`
- By no realm – You can capture all authentication requests that do not contain a realm with the `NoRealm=` command.

Example

```
[Realms]
ABC.com
realm1=myrealm.com
realm2=*abc.com
CatchAllRealm=*
NoRealm=
```

Configuring Each Realm Section

For each realm or alias that you create in the [Realms] section, you must create a separate section identified by the specified realm name or alias in the `gsmmap.gen` file. Within each realm setting, you identify a “target module” for each type of information that might be required to authenticate a subscriber. The target module defines where to obtain the specified information for each type of authenticator.

For example, if `ABC.com` is one of the realms, you must create a target module for any of the SIM, AKA, SMS, IMSI, MSISDN, and Authorization authentication types that will be used to authenticate subscribers from `ABC.com`.

Use the `Default =` setting to identify a target module to be called if any of the other settings are absent.



NOTE: The Setting Name can be set to `None` if you want to disable the setting. For example, `Authorization = None`.

Example

In the following example, these configuration choices are specified:

- An Access-Request from `ABC.com` that requires an SMS text message will be handled according to the settings in the `UlticomSMSGateway` target module section of `gsmmap.gen`.
- Access-Requests requiring an authorization string will be handled according to the settings in the `SQLDatabase` target module section of `gsmmap.gen`.

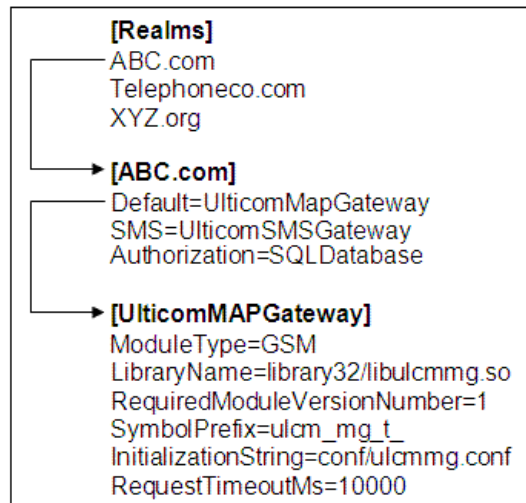
- All other Access-Requests will be handled according to the UlticomMapGateway target module section of `gsmap.gen`.

```
[ABC.com]
Default=UlticomMapGateway
SMS=UlticomSMSSGateway
Authorization=SQLDatabase
```

Relationship Between Sections

Figure 10 illustrates the relationship between the [Realms] section, the specific named realm section, and the target module section in the `gsmap.gen` file.

Figure 10: Relationship Between Sections in `gsmap.gen` File



Network Equipment and Data Needed for Processing Access-Requests

Table 18 identifies the network equipment needed for authentication based on the action needed to process the Access-Request.

Table 18: Network Equipment and Related Settings, Actions, and Identifiers

Setting Name	Action Needed to Process Access-Request	Identifier of the Mobile Station	Network Equipment
SIM	Obtain SIM triplets*	IMSI	HLR (supporting MAP application context version 2)
AKA	Obtain AKA quintets	IMSI	HLR (supporting MAP application context version 3)
SMS	Send SMS text message containing password	IMSI	MSC (SMS text message sent through the MSC)
IMSI	Obtain IMSI (given the MSISDN)	MSISDN	HLR
MSISDN	Obtain MSISDN (given the IMSI)	IMSI	HLR
Authorization	Obtain Authorization string	IMSI or MSISDN	HLR or database

* If quintets are received but triplets are needed, then SIM Server converts the quintets to triplets according to specification 3G TS 33.102 available at <http://www.3gpp.org>.



NOTE: You can set the Setting Name to None if you want to disable the setting. For example, SIM=None.

Example: Authorization String

If an authorization string is required to process an Access-Request, the following might be true:

- Authorization string is in the database
- IMSI is received in the Access-Request
- Database is keyed off the MSISDN

In this case, the MSC is used to obtain the MSISDN based on the IMSI. Then the MSISDN is used to retrieve the Authorization string from the database or HLR.

Disabling Authorization from EAP-SIM

It is possible to disable authorization completely from EAP-SIM (not fetch subscriber profile information from the HLR and not perform a SQL/LDAP query).

To disable authorization from EAP-SIM:

1. Set Authorization=None in the realm section of the `gsmmap.gen` file.

- Remove all authorization options (BS, TS, and ODB) from the `authGateway.conf` file for the target HLR. For more information about `authGateway.conf`, see “Configuring the `ulcmmg.conf` File” on page 52.

Target Module Section

For each target module that you list for a realm, you must create a configuration section that identifies settings to be used for that module. The settings that you must specify depend on the type of module being called. The target modules are described in Table 19.

Table 19: Types of Target Modules

Target Module	Type	Network	Source of Subscriber Information	Default Target Module Name
MAP Gateway	GSM	SS7	HLR	UlticomMapGateway
SMS Gateway	GSM	SS7	MSC	UlticomSMSGateway
SQL Database	Database	IP	SQL database	SQLDatabase
LDAP Database	Database	IP	LDAP database	LDAPDatabase

The fields to be included in the target module section differ depending on the specific target module. For example, the MAP Gateway Target Module section in `gsmmap.gen` requires a different set of fields than the SMS Gateway target module. Table 20 through Table 23 describe the fields that are needed for each target module.

Target Module Fields (General Case)

Table 20: `gsmmap.gen` [Module] Fields (General Case)

Field	Description
ModuleType	Specifies the type of module being called. Options are: <ul style="list-style-type: none"> ■ Database ■ GSM
LibraryName	The name of the library called when the target module runs.
RequiredModuleVersion Number	Version number of the specified module. Default value is 1.
SymbolPrefix	Specifies the prefix for the symbols loaded from the library. <ul style="list-style-type: none"> ■ For the MAP Gateway, enter <code>ulcm_mg_t_</code>. ■ For the SMS Gateway, enter <code>ulcm_sms_t_</code>.
InitializationString	Specifies the name of the initialization file for the library.
RequestTimeoutMs	Specifies the number of milliseconds Steel-Belted Radius waits for a request from the library to complete. The value entered here should reflect how long the SS7 network takes to complete a request. For example, a MAP Gateway communicating with an HLR requires a relatively short timeout value; for example, 10000 (10 seconds). An SMS Gateway that must communicate with a subscriber’s mobile telephone would require a considerably longer timeout value; for example, 60000 (60 seconds).

MAP Gateway Target Module Fields

Table 21: gsmmap.gen [Module] Fields (General case)

Field	Enter...
ModuleType	GSM
LibraryName	library32/libulcmmg.so
RequiredModuleVersion Number	1
SymbolPrefix	ulcm_mg_t_
InitializationString	conf/ulcmmg.conf See “Configuring the ulcmmg.conf File” on page 52 for more information about the ulcmmg.conf file.
RequestTimeoutMs	Number of milliseconds Steel-Belted Radius waits for a request from the library to complete. The value entered here should reflect how long the SS7 network takes to complete a request. For example, a MAP Gateway communicating with an HLR requires a relatively short timeout value; for example, 10000 (10 seconds).

Example of MAP Gateway Target Module Fields

```
[UlticomMAPGateway]
ModuleType=GSM
LibraryName=library32/libulcmmg.so
RequiredModuleVersionNumber=1
SymbolPrefix=ulcm_mg_t_
InitializationString=conf/ulcmmg.conf
RequestTimeoutMs=10000
```

SMS Gateway Target Module Fields

Table 22: gsmmap.gen SMSGateway Fields

gsmmap.gen [Settings] Field	Enter...
ModuleType	GSM
LibraryName	library32/libsmsulcmmg.so
RequiredModuleVersionNumber	1
SymbolPrefix	ulcm_sms_t_
InitializationString	conf/smsulcmmg.conf See “Configuring the SMSGateway.conf File” on page 53 for more information about the smsulcmmg.conf file.
RequestTimeoutMs	Number of milliseconds Steel-Belted Radius waits for a request from the library to complete. The value entered here should reflect how long the SS7 network takes to complete a request. For example, an SMS Gateway that must communicate with a subscriber’s mobile telephone would require a relatively long timeout value; for example, 60000 (60 seconds).

Example of SMS Gateway Target Module

```
[UlticomSMSTGateway]
ModuleType=GSM
LibraryName=library32/libsmsulcmmg.so
RequiredModuleVersionNumber=1
SymbolPrefix=ulcm_sms_t_
InitializationString=conf/smsulcmmg.conf
RequestTimeoutMs=60000
```

SQL Database Target Module Fields

Table 23: gsmmap.gen SQL Database Fields

gsmmap.gen [Database] Field	Enter...
ModuleType	Database
DatabaseAccessor MethodName	Name by which the SQL data accessor registers itself with Steel-Belted Radius. This value must match the value entered in the MethodName setting in the <code>sqlaccessor.gen</code> file (described in “Configuring sqlaccessor.gen and sqlaccessorjdbc.gen” on page 67).
KeyForAuthorization	Specifies whether the subscriber will be identified by IMSI or MSISDN (key field). Valid values are: <ul style="list-style-type: none"> ■ IMSI ■ MSISDN For more information about setting database keys, see “Identifying Key Fields for Oracle, JDBC, and LDAP Databases” on page 82.

Example of SQL Database Target Module

```
[SQLDatabase]
ModuleType=Database
DatabaseAccessorMethodName=SQL Accessor
KeyForAuthorization=MSISDN
```

LDAP Database Target Module Fields

Table 24: gsmmap.gen Database Fields

Field	Description
ModuleType	Database
DatabaseAccessor MethodName	Name by which the SQL data accessor registers itself with Steel-Belted Radius. This value must match the value entered in the MethodName setting in the <code>ldapaccessor.gen</code> file (described in “Configuring the LDAP Data Accessor (ldapaccessor.gen)” on page 76).

Table 24: gsmmap.gen Database Fields

Field	Description
KeyForAuthorization	<p>Specifies whether the subscriber will be identified by IMSI or MSISDN. Valid values are:</p> <ul style="list-style-type: none"> ■ IMSI ■ MSISDN <p>For more information about setting database keys, see “Identifying Key Fields for Oracle, JDBC, and LDAP Databases” on page 82.</p>

Example of LDAP Database Target Module

```
[LDAPDatabase]
ModuleType=Database
DatabaseAccessorMethodName=LDAP Accessor
KeyForAuthorization=IMSI
```

Configuring sqlaccessor.gen and sqlaccessorjdbc.gen

You can use an external LDAP directory or SQL database to authorize subscribers. The `sqlaccessor.gen` file stores the settings needed by the `SQLAccessor` plugin to authorize subscribers. `SQLAccessor` requires three items of information from the database:

- IMSI
- MSISDN
- Authorization String

This section describes the configuration choices that you need to make to configure `sqlaccessor.gen` or `sqlaccessorjdbc.gen`.

The `sqlaccessor.gen` file stores the settings used by the SQL data accessor plug-in. It is composed of several sections. Section names are enclosed in square brackets.



NOTE: Databases should support stored procedures.

[Bootstrap] Section

The [Bootstrap] section of the `sqlaccessor.gen` file contains the following settings:

Table 25: sqlaccessor.gen [Bootstrap] Fields

Field	Description
LibraryName	Specifies the name of the library called when <code>gsmmap</code> is loaded by SIM Server. Default value is <code>radsql_accessor_ora8</code> .
Enable	Set to 1 to enable SQL access. Set to 0 to disable SQL access. Default value is 0.

Oracle Example

```
[Bootstrap]
LibraryName=radsql_accessor_ora8
Enable=0
```

JDBC Example

```
[Bootstrap]
LibraryName=radsql_accessor_jdbc
Enable=0
```

[Settings] Section

The [Settings] section of the `sqlaccessor.gen` file defines parameters that control the database connection.

Table 26: sqlaccessor.gen and sqlaccessorjdbc.gen [Settings] Fields

Field	Description
MethodName	Identifies the name under which the data accessor registers itself with Steel-Belted Radius. Default value is <code>SQL Accessor</code> .
Connect	Specifies the string that must be passed to the database client engine to establish a connection to the database. This string has (or refers to) information about the name of the database, its location on the network, the password required to access it, and so forth.
SQL	Contains the SQL statement used to access the subscriber authentication information in the database. The SQL statement can be broken over several lines by ending each line with a backslash. The backslash must be preceded by a space character, and followed by a newline character. You can indent the subsequent lines for better readability.
ParameterMarker	Specifies the character or sequence of characters used as the parameter marker in a parameterized SQL query. Default value is <code>?</code> .
MaxConcurrent	Specifies the maximum number of instances of a single SQL statement that can be executing at one time. Default value is 1.

Table 26: sqlaccessor.gen and sqlaccessorjdbc.gen [Settings] Fields (continued)

Field	Description
ConcurrentTimeout	Specifies the number of seconds that a request waits for execution before it is discarded. Because as many as <i>MaxConcurrent</i> SQL statements can be executing at one time, new requests must be queued as they arrive until other statements are processed.
ConnectTimeout	Specifies the number of seconds to wait when attempting to establish the connection to the database before timing out. This value is passed to the client database engine, which might or might not implement the feature. Default value is 25 seconds.
QueryTimeout	Specifies the number of seconds to wait for a response to a query before timing out. This value is passed to the client database engine, which might or might not implement the feature. Default value is 25 seconds.
WaitReconnect	Specifies the number of seconds to wait after a failure of the database connection before trying to connect again. Default value is 2 seconds.
MaxWaitReconnect	Specifies the maximum number of seconds to wait after successive failures to reconnect after a failure of the database connection. The WaitReconnect setting specifies the time to wait after failure of the database connection. This value is doubled on each failed attempt to reconnect, up to the value of the number of seconds specified by the MaxWaitReconnect setting. Default value is 360 seconds (6 minutes).

Oracle Example: sqlaccessor.gen [Settings] section

In the following example, the `Connect=` statement in the example establishes the connection to the database. The `SQL=` statement in the example queries the database, selecting the values for the `msisdn`, `imsi`, and authorization string. The selection from the database is based on the key field. Note that the statement `WHERE subscriber_id=@KeyToRecord` in the example indicates that the `subscriber_id` field in the example database is the key field.

For a description of each field in the [Settings] section, see Figure 26 on page 68. For an annotated version of the `sqlaccessor.gen` file including instructions, see the `sqlaccessor.gen` file on the CD.

```
[Settings]
MethodName=SQL Accessor
Connect=username/password@servicename
SQL=SELECT msisdn, imsi, auth_string FROM tablename WHERE subscriber_id=@KeytoRecord
ParameterMarker=?
MaxConcurrent=1
ConcurrentTimeout=30
ConnectTimeout=25
QueryTimeout=25
WaitReconnect=2
MaxWaitReconnect=360
```

JDBC Example: sqlaccessorjdbc.gen [Settings] section

In the following example, the `Connect=` statement in the example establishes the connection to the database. The `SQL=` statement in the example queries the database, selecting the values for the `msisdn`, `imsi`, and authorization string. The selection from the database is based on the key field. Note that the statement in the example `WHERE subscriber_id=@KeyToRecord` indicates that the `subscriber_id` field in the example database is the key field.

For a description of each field in the [Settings] section, see Figure 26 on page 68. For an annotated version of the `sqlaccessorjdbc.gen` file including instructions, see the `sqlaccessorjdbc.gen` file on the CD.

```
[Settings]
MethodName=SQL Accessor
Driver=com/provider/jdbc/sqlserver/SQLServerDriver
ConnectDelimiter=;
Connect=DSN=jdbc:provider:driver1:dsn_name;UID=db_username;PWD=db_password
SQL=SELECT msisdn, imsi, auth_string FROM tablename WHERE subscriber_id=@KeyToRecord
ParameterMarker=?
MaxConcurrent=1
ConcurrentTimeout=30
ConnectTimeout=25
QueryTimeout=25
WaitReconnect=2
MaxWaitReconnect=360
```

SQL Server Connection Methods

There are two methods to establish a connection to the server:

- Connect to a single SQL server
For more information, see [Connecting to a Single SQL Server](#).
- Connect to multiple SQL servers
For more information, see “Connecting to Multiple Servers” on page 71.

Connecting to a Single SQL Server

To connect to a single server, include a connect statement in the [Settings] section of `sqlaccessor.gen` or `sqlaccessorjdbc.gen`. For example:

```
[Settings]
MethodName=SQL Accessor
Connect=username/password@servicename
:
```

Connecting to Multiple Servers

[Server] Section

Steel-Belted Radius can maintain multiple SQL server connections and authenticate users against authentication databases in a round-robin fashion. The [Server] section of the `sqlaccessor.gen` file enables you to distribute the authentication workload across several servers by giving Steel-Belted Radius a pool of servers from which to create the round-robin list. The [Server] section identifies each server that you can use.

The syntax is as follows:

```
[Server]
ServerName=TargetNumber
ServerName=TargetNumber
:
```

Table 27: sqlaccessor.gen File [Server] Fields

Field	Description
ServerName	The name of the <code>sqlaccessor.gen</code> file section that contains configuration information for that server.
TargetNumber	An <i>activation target number</i> , a number that controls when this server is activated for backup purposes. <i>TargetNumber</i> is optional and can be left blank. An activation target value of 0 indicates that, in the current configuration, this machine is never used.

A Steel-Belted Radius server maintains connectivity with its SQL servers according to the following rules:

- The priority of the server by order. The first entry in the [Server] section has the highest priority.
- By activation target number. The rule for the activation target is that if the number of SQL servers to which Steel-Belted Radius is connected is less than the activation target, Steel-Belted Radius connects to the server and includes it in the round-robin list. While the number of active servers is equal to or greater than the activation target, Steel-Belted Radius does not use that server in the round-robin list.

[Server/name] Sections

You must provide a [Server/*name*] section for each server that you named in the [Server] section of the `sqlaccessor.gen` file. Each [Server/*name*] section must identify the database that you want to use and the user name and password required to access that database.

```
[Server/name]
Connect=username/password@servicename
```

You can identify a “last resort” SQL server by providing a `LastResort` field in one of these `[Server/name]` sections, and setting its value to 1. If a SQL query against some other server results in “no record found,” the authentication server tries the last resort server before accepting or rejecting the user.

In the following example, server `s3` is the last resort server; the `@mydb` string refers to the service name for an Oracle database in the `tnsnames.ora` file (the server won’t connect to the Oracle database without this).

```
[Server]
s1=2
s2=2
s3=1

[Server/s1]
Connect=system1/manager

[Server/s2]
Connect=system2/manager@mydb2

[Server/s3]
Connect=system3/manager@mydb3
LastResort = 1
```

You might use the `LastResort` field to identify your master accounts database. This enables Steel-Belted Radius to authenticate the user in the case where a user account is newly added to the master accounts database but has not yet been propagated to all the SQL databases.

SQL Database Data Retrieval Methods

There are two methods for retrieving the required data items (IMSI, MSISDN, and Authorization String) from the SQL database:

- **SQL=SELECT Statement Method**
For specific information see `SQL=SELECT Method for Data Retrieval from SQL Databases`.
- **Stored Procedure Method**
For specific information see “Stored Procedure Method for Data Retrieval from SQL Databases” on page 74.

SQL=SELECT Method for Data Retrieval from SQL Databases

The `SQL=SELECT` method for retrieving the IMSI, MSISDN, and Authorization String from the SQL database involves including a `SQL=SELECT` statement and a corresponding `[Results]` section in the `sqlaccessor.gen` or `sqlaccessorjdbc.gen` file.

SQL=SELECT Method: SELECT Statement

Place a `SQL=SELECT` Statement in the `[Settings]` section of `sqlaccessor.gen` or `sqlaccessorjdbc.gen` to retrieve the IMSI, MSISDN, and Authorization String.

In the following example, the IMSI, MSISDN, and Authorization String are selected from a subscriber database in which the MSISDN is the key. (The `user` column contains MSISDN values.)

```
SQL = SELECT user, msisdn, authstring FROM my_database WHERE user=@KeyToRecord
```

SQL=SELECT Method: [Results] Section

The [Results] section maps the position of a column name in the SELECT SQL statement with the data needed. In the following example, the [Results] section indicates that the first item in the SQL = SELECT statement names the database column in which the imsi is found and that the column is 16 characters wide.

```
[Results]
ResultIMSI = 1/16
ResultMSISDN = 2/16
ResultAuthString = 3/16
```

SQL=SELECT Method: Section Correlations Illustrated

Figure 11 on page 74 illustrates the correlation between the SQL=SELECT statement, the [Results] section, the SQL database, and the key identified in `gsmmap.gen`.

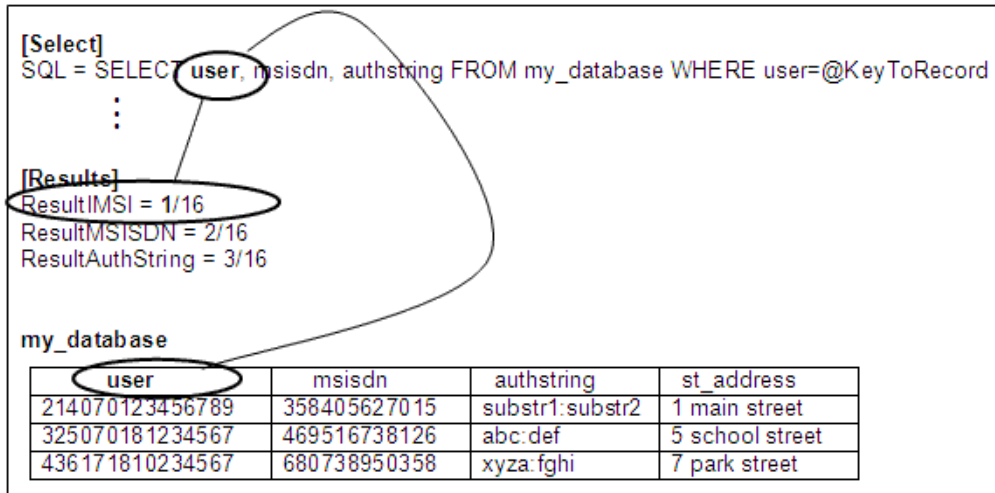
`ResultMSISDN=1/16` indicates that position 1 in the SQL=SELECT statement contains the column heading of the database from which the value for `ResultIMSI` will be retrieved. In Figure 11 on page 74, the column heading containing IMSI values is `user`.

Note that the column headings are not required to be `imsi`, `msisdn`, or `authstring`. The column headings, however, must map to `ResultIMSI`, `ResultMSISDN`, and `ResultAuthString` as shown in Figure 11 on page 74. For example, `imsi` in the SQL=SELECT statement maps to the column named `user`.

In Figure 11 on page 74, the line `KeyForAuthorization=IMSI` in the `gsmmap.gen` file indicates that the column that is used as the key contains IMSI values. The `sqlaccessor.gen` file indicates that the `user` column is the key (`WHERE user=@KeyToRecord`). Therefore, the `user` column must contain IMSI values.

Figure 11: Relationship Between Sections in sqlaccessor.gen File**gsmmap.gen**

```
[SQLDatabase]
ModuleType=Database
DatabaseAccessorMethodName=SQL Accessor
KeyForAuthorization=IMSI
:
```

sqlaccessor.gen**Stored Procedure Method for Data Retrieval from SQL Databases**

You can use a stored procedure, rather than a SQL=SELECT statement, to retrieve the IMSI, MSISDN, and Authorization String from the database for use by the SQLAccessor plugin.

The stored procedure must be created in the Oracle database before using it in SIM Server.

The simauth and smsprov plug-ins expect the IMSI, MSISDN, and Authorization String in the format used by the MAP Gateway. However, the SQL database schema might not allow these strings to be obtained in the expected format. Therefore, the SQLAccessor module can use a stored procedure to convert the database information to the expected format.

Stored Procedure Method: BEGIN Statement of sqlaccessor.gen

Include a SQL=BEGIN statement in the [Settings] section of sqlaccessor.gen or sqlaccessorjdbc.gen to convert the data from the database to the output parameters, ResultIMSI, ResultMSISDN, and ResultAuthString.

Example:

```
SQL=BEGIN SIM_Server_stored_proc.produce_return_vals(@KeyToRecord!i,
@ResultIMSI!o, @ResultMSISDN!o, @ResultAuthString!o); END;
```

Stored Procedure Method: [Results] Section of sqlaccessor.gen

The stored procedure converts and maps SQL values to the variables listed in the [Results] section. If you are using the stored procedure method, include this [Results] section in sqlaccessor.gen or sqlaccessorjdbc.gen exactly as shown here.

```
[Results]
ResultIMSI
ResultMSISDN
ResultAuthString
```

Stored Procedure Method: Database Schema

The database schema must exist for the database key and the data to be retrieved from the database.

Example:

```
msisdn VARCHAR2(32)
user VARCHAR2(32)
authstring VARCHAR2(32)
```

(Note that the column named `user` contains IMSI values.)

Stored Procedure Method: Data Retrieval

The stored procedure must retrieve the values for the IMSI, MSISDN, and Authorization String from the database and return them in the values of `ResultIMSI`, `ResultMSISDN`, and `ResultAuthString`.

Stored Procedure Method: Example

The following lines retrieve the values for IMSI, MSISDN, and Authorization String and place them in the output parameters `ResultIMSI`, `ResultMSISDN`, and `ResultAuthString`. Note that the database columns are named `msisdn`, `authstring`, and `user`, where `user` contains IMSI values. The IMSI values are the key values.

```

      :
CREATE OR REPLACE PACKAGE SIM_Server_stored_proc IS
  PROCEDURE produce_return_vals(
KeyToRecord    IN VARCHAR2,
ResultIMSI     OUT VARCHAR2,
ResultMSISDN  OUT VARCHAR2,
ResultAuthString OUT VARCHAR2) IS
      :

```

– The cursor holds the result of the query.

```
CURSOR cur IS
SELECT * FROM my_database WHERE user=KeyToRecord;
. – Execute the query
```

```

OPEN cur;
FETCH cur INTO row;
  ⋮

```

– If the row was found then convert the data

```

IF( cur%FOUND ) THEN

ResultMSISDN := row.msisdn;
ResultIMSI := row.user;
ResultAuthString:= row.authstring;
  ⋮

```

Configuring the LDAP Data Accessor (ldapaccessor.gen)

The `ldapaccessor.gen` file stores the settings used by the LDAP data accessor plug-in. The `ldapaccessor.gen` file is composed of several sections. Section names are enclosed in square brackets.

[Bootstrap] Section

The `[Bootstrap]` section of the `ldapaccessor.gen` file contains the following settings:

```

[Bootstrap]
LibraryName=ldapaccessor
Enable=0

```

Table 28: ldapaccessor.gen [Bootstrap] Fields

Field	Description
LibraryName	Specifies the name of the library called when <code>gsmmap</code> runs. Default value is <code>ldapaccessor</code> .
Enable	Set to 1 to enable LDAP access. Set to 0 to disable LDAP access. Default value is 0.

[Settings] Section

The `[Settings]` section of the `ldapaccessor.gen` file defines parameters that control the database connection.

```

[Settings]
MethodName=LDAP Accessor
MaxConcurrent=1
Timeout=20
ConnectTimeout=25
QueryTimeout=25
WaitReconnect=2
MaxWaitReconnect=360
; BindName=uid=<User-Name>, ou=sales, o=bigco.com
LogLevel = 2
UpperCaseName = 0

```

PasswordCase=original
 PasswordFormat = 0
 Search = DoLdapSearch
 SSL = 0
 MaxScriptSteps = 50

Table 29: Idapaccessor.gen [Settings] Fields

Field	Description
MethodName	<p>Identifies the name under which the data accessor registers itself with Steel-Belted Radius.</p> <p>Default value is LDAP Accessor.</p>
MaxConcurrent	<p>Specifies the maximum number of instances of a single LDAP statement that can be executing at one time.</p> <p>Default value is 1.</p>
Timeout	<p>Specifies the number of seconds that a request waits for execution before it is discarded.</p> <p>Because as many as MaxConcurrent LDAP statements can be executing at one time, new requests must be queued as they arrive until other statements are processed.</p>
ConnectTimeout	<p>Specifies the number of seconds to wait when attempting to establish the connection to the LDAP directory before timing out. This value is passed to the client LDAP directory, which might or might not implement the feature.</p> <p>Default value is 25 seconds.</p>
QueryTimeout	<p>Specifies the number of seconds to wait for a response to a query before timing out. This value is passed to the client LDAP directory, which might or might not implement the feature.</p> <p>Default value is 10 seconds.</p>
WaitReconnect	<p>Specifies the number of seconds to wait after a failure of the LDAP directory connection before trying to connect again.</p> <p>Default value is 2 seconds.</p>
MaxWaitReconnect	<p>Specifies the maximum number of seconds to wait after successive failures to reconnect after a failure of the LDAP directory connection.</p> <p>The WaitReconnect setting specifies the time to wait after failure of the LDAP directory connection. This value is doubled on each failed attempt to reconnect, up to the value of the number of seconds specified by the MaxWaitReconnect setting.</p> <p>Default value is 360 seconds (6 minutes).</p>
BindName	<p>Specifies the default distinguished name (DN) to be used in the Bind request that connects to LDAP servers.</p> <p>The [Server/name] section enables you to specify a unique BindName for a specific server. Use the [Settings] section to specify a default BindName to use for all servers.</p>

Table 29: ldapaccessor.gen [Settings] Fields (continued)

Field	Description
LogLevel	<p>Activates logging for the LDAP authentication component and sets the rate at which it writes entries to the Steel-Belted Radius server activity log file (.LOG). This value can be the number 0, 1, or 2, where 0 is the lowest logging level, 1 is intermediate, and 2 is the most verbose. The LogLevel is re-read whenever the server receives a HUP signal.</p> <p>If the LogLevel that you set in the .aut file is different than the LogLevel in radius.ini, the radius.ini setting determines the rate of logging.</p>
UpperCaseName	<p>Specifies whether the username should be converted to uppercase. Choices are: 0 (preserve the case of the username), 1 (convert username to uppercase).</p> <p>Default value is 0.</p>
PasswordCase	<p>If set to U or Upper, the password returned from the LDAP database is converted to uppercase before authentication. If L or Lower, the password is converted to lowercase. If O or Original, the password is not altered before authentication.</p> <p>Default value is Original.</p>
PasswordFormat	<p>By default, the PasswordFormat parameter is not listed in the [Settings] section of the LDAP authentication header file. With no listing, Steel-Belted Radius expects the user's password in the LDAP table to be in clear text format.</p> <p>If you want to configure Steel-Belted Radius to automatically handle password values correctly when it detects that they have been encrypted using UNIXcrypt or a SHA1 + Base64 hash, then set PasswordFormat to auto.</p>
Search	<p>The value of this field is a string, <i>name</i>. The <i>name</i> specifies an LDAP Search request by referencing a [Search/<i>name</i>] section elsewhere in the file.</p>
SSL	<p>Specifies whether to use SSL over the LDAP connection. The choices are: 0 (do not use SSL), 1 (use SSL).</p> <p>Default value is 0.</p>
MaxScriptSteps	<p>Specifies the maximum number of statements a script can execute before terminating. You can use the MaxScriptSteps parameter to ensure that a script does not enter an infinite loop condition.</p>

[Server] Section

Steel-Belted Radius can maintain multiple LDAP server connections and authenticate users against authentication databases in a round-robin fashion. The [Server] section of the ldapaccessor.gen file enables you to distribute the authentication workload across several servers by giving Steel-Belted Radius a pool of servers from which to create the round-robin list. The [Server] section identifies each server that you can use.

The syntax is as follows:

```
[Server]
ServerName=TargetNumber
ServerName=TargetNumber
⋮
```

Table 30: Idapaccessor.gen File [Server] Fields

Field	Description
ServerName	The name of the <code>Idapaccessor.gen</code> file section that contains configuration information for that server.
TargetNumber	An <i>activation target number</i> , a number that controls when this server is activated for backup purposes. <i>TargetNumber</i> is optional and can be left blank. An activation target value of 0 indicates that, in the current configuration, this machine is never used.

A Steel-Belted Radius server maintains connectivity with its LDAP servers according to the following rules:

- Priority of the server by order. The first entry in the [Server] section has the highest priority.
- Activation target number. The rule for the activation target is that if the number of LDAP servers to which Steel-Belted Radius is connected is less than the activation target, Steel-Belted Radius connects to the server and includes it in the round-robin list. While the number of active servers is equal to or greater than the activation target, Steel-Belted Radius does not use that server in the round-robin list.

[Server/name] Sections

You must provide a [Server/*name*] section for each server that you named in the [Server] section of the `Idapaccessor.gen` file. Each [Server/*name*] section must identify the database that you want to use and the user name and password required to access that database.

```
[Server]
s1=
```

```
[Server/s1]
LdapVersion=3
Host=LDAPServerNameOrIPAddress
Port = 389
BindName=uid=admin, ou=sales, o=bigco.com
BindPassword=secret
```

[Request] Section

You must use the [Request] section of `ldapaccessor.gen` to bind the `KeyToRecord` variable provided by the `gsmmap` module to the `Key` variable used in the LDAP search definitions. Note that the value specified here (**Key**) must match the value specified in the [Search/DoLdapSearch] section.

See “Identifying Key Fields for Oracle, JDBC, and LDAP Databases” on page 82 for more information about key fields.

```
[Request]
KeyToRecord = Key
```



NOTE: Do not modify the `KeyToRecord` keyword in the [Request] section. The value `KeyToRecord` is hard-coded into the `gsmmap` module.

[Response] Section

The [Response] section of the `ldapaccessor.gen` file maps the information retrieved by the LDAP search to values expected by the `gsmmap` module. Do not change the [Response] section unless instructed to do so by Juniper Networks Technical Services.

```
[Response]
ResultIMSI =
ResultMSISDN =
ResultAuthString =
```

[Attributes/AttrList] Section

The [Attributes/AttrList] section identifies the attributes contained in the LDAP schema. Replace the attribute names in the sample file with the attributes used at your site.

```
[Attributes/AttrList]

wlanMSISDN
wlanIMSI
wlanAuthorization
wlanPrepayFlag
```

[Script] Section

The [Script] section of `ldapaccessor.gen` contains an example Javascript routine that controls how the information returned from the LDAP directory (which must consist of values for `ResultIMSI`, `ResultMSISDN`, and `ResultAuthString`) to the format required by the `gsmmap` module.

```
[Script]
//
// Clear the output variables
LdapVariables.Reset("ResultMSISDN");
LdapVariables.Reset("ResultIMSI");
LdapVariables.Reset("ResultAuthString");
```

```

// Search the database
status = Ldap.Search('DoLdapSearch');

// If the record was not found then return
if(status==Ldap.NOTFOUND)
{
    return SBR_RET_FAILURE;
}

// Pass-through the MSISDN and IMSI obtained from the LDAP directory:
var _msisdn = LdapVariables.Get("wlanMSISDN");
LdapVariables.Add("ResultMSISDN", _msisdn);
var _imsi = LdapVariables.Get("wlanIMSI");
LdapVariables.Add("ResultIMSI", _imsi);

// Values of wlanAuthorization and wlanPrepayFlag are "TRUE" or "FALSE".
var _auth = LdapVariables.Get("wlanAuthorization");
var _prepay = LdapVariables.Get("wlanPrepayFlag");

// Construct the authorization string. The format is "substr1:substr2:substr3:".
// (The final colon is optional.)
var _authstring = "";
if(_auth == "TRUE")
{
    // Add the substring "WLANAuthorized" to the string.
    _authstring += "WLANAuthorized:";
}

if(_prepay == "FALSE")
{
    // Add the substring "NotPrepay:" to the string.
    _authstring += "NotPrepay:";
}
LdapVariables.Add("ResultAuthString", _authstring);

return SBR_RET_SUCCESS;
// End of script

```

[Search/DoLdapSearch] Section

The [Search/DoLdapSearch] section specifies the values for Base and Scope for the Javascript routine in the [Script] section of the `gsmmap.gen` file. The [Search/DoLdapSearch] section also sets up a filter that identifies the key to the LDAP database.

For more information about setting database keys, see “Identifying Key Fields for Oracle, JDBC, and LDAP Databases” on page 82.

```
[Search/DoLdapSearch]
;
; Base = o=bigco.com
Base = o=sbrsms,c=US
Scope = 2

Filter = wlanMSISDN=<Key>
Attributes = AttrList
Timeout = 20
%DN = dn
```

Identifying Key Fields for Oracle, JDBC, and LDAP Databases

You can use the IMSI or the MSISDN as the key field for retrieving subscriber information from a SQL database or LDAP directory. You need to configure the following files to identify and use the MSISDN or IMSI as the key field:

Table 31: Files to be Configured when Identifying the Key Field

Oracle or JDBC (SQL)	LDAP
gsmmap.gen [SQLDatabase] section	gsmmap.gen [LDAPDatabase] section
sqlaccessor.gen or sqlaccessorjdbc.gen [Settings] section	ldapaccessor.gen [Request] section and [Search/DoLdapSearch] section
oracle or jdbc database	LDAP directory

Configuring gsmmap.gen for Key Field Identification

The choice of IMSI or MSISDN as the key field is identified in the gsmmap.gen file with the KeyForAuthorization field. (KeyForAuthorization can be MSISDN or IMSI.) In the following examples, MSISDN is identified as the key field.

Examples

gsmmap.gen file (Oracle or JDBC)

```
[SQLDatabase]
ModuleType=Database
DatabaseAccessorMethodName=SQL Accessor
KeyForAuthorization=MSISDN
```

gsmmap.gen file (LDAP)

```
[LDAPDatabase]
ModuleType=Database
DatabaseAccessorMethodName=LDAP Accessor
KeyForAuthorization=MSISDN
```

Configuring sqlaccessor.gen or sqlaccessorjdbc.gen for Key Field Identification

For SQL databases, the SELECT statement in the [Settings] section of sqlaccessor.gen or sqlaccessorjdbc.gen identifies the database column name of the key field. In the following example, subscriber_id is identified as the column containing the key field.

Examples

Oracle Example: sqlaccessor.gen file

```
[Settings]
MethodName=SQL Accessor
Connect=my_user_name/password@servicename
SQL=SELECT service_type FROM table1 WHERE subscriber_id=@KeyToRecord
```

JDBC Example: sqlaccessorjdbc.gen file

```
[Settings]
MethodName=SQL Accessor
Driver=com/provider/jdbc/sqlserver/SQLServerDriver
ConnectDelimiter=;
Connect=DSN=jdbc:provider:driver1:dsn_name;UID=db_username;PWD=db_password
SQL=SELECT service_type FROM table1 WHERE subscriber_id=@KeyToRecord
```

The corresponding database must contain a column name (as specified in the SELECT statement) containing the key field. In the following example, the column name of `subscriber_id` contains MSISDN values that serve as record keys.

Example SQL Database (The `subscriber_id` column contains the key MSISDN data)

subscriber_id	service_type	street_address
1234	basic	10 Main Street
6889	premium	15 School Street

Configuring `ldapaccessor.gen` for Key Field Identification

For LDAP directories, the [Request] section and the [Search/DoLdapSearch] section identifies the key to the LDAP directory. You can set the `KeyToRecord` field in the [Request] section to either MSISDN or IMSI. The [Search/DoLdapSearch] section identifies the column name in the LDAP directory that contains the record key.

ldapaccessor.gen file

```
[Request]
KeyToRecord=key
:
[Search/DoLdapSearch]
Base=o=bigco.com
Scope=2
:
Filter=wlanMSISDN=key
```



NOTE: See “Configuring the LDAP Data Accessor (`ldapaccessor.gen`)” beginning on page 76 for more information about `ldapaccessor.gen`. See “[Script] Section” on page 80 for an example LDAP script.

Example LDAP Directory (The `wlanMSISDN` column contains the key MSISDN data)

wlanMSISDN	service_type	street_address
1234	basic	10 Main Street
6889	premium	15 School Street

Chapter 5

Special Attribute Handling

This chapter describes configuration tasks for special attribute handling features including:

- Adding Location Information to Access-Requests — Adds NAS location information attributes to an Access-Request
- Assigning IP Addresses Based on Access Point Name (APN) — Enables Steel-Belted Radius to assign an IP address for the mobile node based on the Access Point.
- Adding Attributes to an Access-Accept — Assigns a value from the subscriber database to an attribute and returns it with the Access-Accept.

Adding Location Information to Access-Requests

Service providers might require the location of the mobile device that is requesting access. For example, a service provider might offer weather reports or advertising based on the location of the mobile device.

You can configure an Access-Request to include the location of the NAS through which the proxied request was processed. The NAS is geographically near the mobile device. The location of the NAS closely approximates the location of the mobile device.

Overview

When a mobile device is outside the area of its provider, it roams by sending the request to a local foreign AAA (FAAA) server that is owned by another provider. The FAAA server proxies (forwards) the request to the appropriate home AAA (HAAA) server for the user.

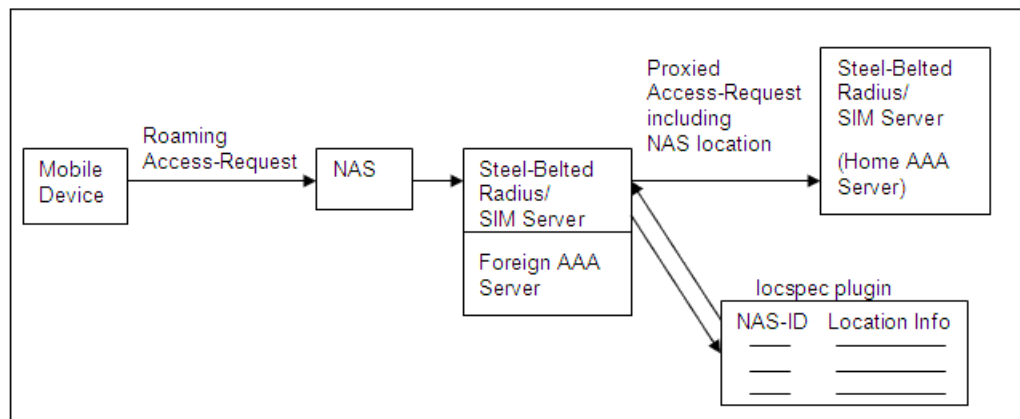
For proxied requests, Steel-Belted Radius can perform a lookup to find a NAS location based on an attribute (usually NAS-Identifier or NAS-IP-Address). The attribute that is used to look up the NAS location is user-configurable as the `AttributeToIdentifyNAS` in the `locspec.ctrl` file. The lookup information is user-configured in the `locspec.ctrl` file.

Figure 12 shows that Steel-Belted Radius queries the `locspec` plugin to find the value of the attribute that identifies the NAS location. The NAS location is then added to the Access-Request that is sent to the service provider's Home AAA server.



NOTE: Each Steel-Belted Radius server that might be the target of a proxy request must be set up as a proxy target. Set up proxy targets with the Steel-Belted Radius Administrator. See the *Steel-Belted Radius Administration Guide* for more information about proxy targets.

Figure 12: Addition of NAS Location to Access-Request



Location-Specific Configuration Files

The following files and file sections require configuration to add location information to the Access-Request. Figure 14 on page 91 provides an example showing the relationship between all the configuration files.

locspec.ctrl file

- [Bootstrap] section
- [Settings] section
- [NAS-LIST] section
- [NAS Identifier] section

proxy.ini file

- [Realms] section

realm.pro file

- [Auth-Outbound-To-Proxy] section
- [Acct-Outbound-To-Proxy] section



NOTE: See the *Steel-Belted Radius Administration Guide* and the *Steel-Belted Radius Reference Guide* for complete information about configuring these files for functionality other than location-specific information.

locspec.ctl File

The `locspec.ctl` file calls the `LOCSPEC` control point plug-in, which enables the addition of location-specific information to an Access-Request.

Table 32 defines the fields needed in the [Bootstrap] section for adding location-specific attributes to an Access-Request.

Table 32: locspec.ctl [Bootstrap] Fields

Field	Description
LibraryName	Specifies the name of the library called. Set to <code>locspec.so</code>
Enable	Set to 1 to enable this file. Set to 0 to disable this file. Set to 1.
InitializationString	Specifies the name of the control point plug-in file that activates location-specific information. Set to <code>LOCSPEC</code> .

Example

```
[Bootstrap]
LibraryName=locspec.so
Enable=1
Initializationstring=LOCSPEC
```

Table 33 defines the fields needed in the [Settings] section for adding location-specific attributes to an Access-Request.

Table 33: locspec.ctl [Settings] Fields

Field	Description
AttributeToIdentifyNAS	Attribute to be used to identify the NAS. Typically, this value is set to one of the following: NAS-Identifier NAS-IP-Address
ConfigLog	Method for capturing log information. ConfigLog = None means that configuration information will not be captured. ConfigLog = ConsoleAndLog sends the log information to both the console and the log. ConfigLog = Console sends the log information to the log file only. ConfigLog = Log sends the log information to the log file only. Default is ConsoleandLog.

Example

```
[Settings]
AttributetoidentifyNAS=NAS-Identifier
ConfigLog=ConsoleAndLog
```

Table 34 defines the fields needed in the [NAS-LIST] section for adding location-specific attributes to an Access-Request.

Table 34: locspec.ctrl [NAS-LIST] Fields for Configuration of Location-specific Attributes

Field	Description
<i>NAS designator</i>	<p>List of NAS devices.</p> <p>The [NAS-List] section includes a list of NAS devices that are being configured to transmit their location. The attribute used to identify a NAS in this list is configured in the AttributeToIdentifyNAS field within the [Settings] section of the locspec.ctrl field. Typically, the NAS-Identifier attribute or the NAS-IP-Address attribute is used to identify a NAS.</p> <p>For example, if AttributeToIdentifyNAS = NAS-IP-Address, then all the NAS devices in this list will be identified by their IP Address. If AttributeToIdentifyNAS = NAS-Identifier, then all the NAS devices in this list will be identified by their NAS Identifier (name).</p>

Example

```
[NAS-LIST]
NAS_1
NAS_2
```

For each NAS device listed in the [NAS-LIST] section, there must be a separate section in locspec.ctrl providing location information about the NAS.

Table 35 defines the fields needed in the [NAS Identifier] section that provide location-specific information to an Access-Request. The Access-Request can contain all of these four attributes or a subset.

Table 35: Location Attributes for the NAS Device

Field	Description
GSM-Operator-Name	<p>GSM-Operator-Name = <i>prefix:value</i></p> <p>where</p> <p><i>prefix</i> = either GSM or REALM</p> <p><i>code</i> = If <i>prefix</i> = GSM, <i>code</i> = any GSMA assigned TADIG code in capital ASCII letters available at http://www.gsmworld.org; If <i>prefix</i> = REALM, <i>code</i> = or any valid domain name string</p>
GSM-Location-Information	<p>GSM-Location-Information = country=<i>code</i>[:<i>civic-label=value</i>]</p> <p>where</p> <p><i>code</i> = ISO 3166 2-letter country code.</p> <p><i>civic-label</i> = A1, A2, A3, A4, A5, A6, PRD, POD, STS, HNO, HNS, LMK, LOC, NAM, ZIP, PCN, or an integer as defined in draft-ietf-geopriv-dhcp-civil-09.txt available at http://www.ietf.org.</p>
GSM-Visited-Operator-Id	<p>GSM-Visited-Operator-Id = <i>prefix:value</i></p> <p>where</p> <p><i>prefix</i> = either TADIG or REALM</p> <p><i>code</i> = If <i>prefix</i> = GSM, <i>code</i> = any GSMA assigned TADIG code in capital ASCII letters available at http://www.gsmworld.org; If <i>prefix</i> = REALM, <i>code</i> = or any valid domain name string</p>

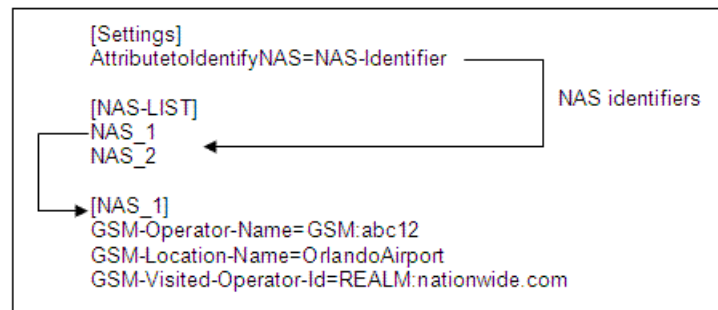
Table 35: Location Attributes for the NAS Device (continued)

Field	Description
GSM-Location-Name	GSM-Location-Name = <i>value</i> where <i>value</i> = textual description of the WLAN Hot Spot (human readable string without mandated format).

Example

```
[NAS_1]
GSM-Operator-Name=REALM:worldnetwork.com
GSM-Location-Name=BostonNeighborsClub
GSM-Visited-Operator-Id=GSM:USACD
GSM-Location-Information=country=US;A1=MA;A3=Boston;ZIP=02116
```

Figure 13 shows the relationship between the `AttributetoidentifyNAS` setting, the `NAS-LIST` section, and the `NAS identifier` section of the `locspec.ctrl` file.

Figure 13: [Settings], [NAS-LIST], and [NAS identifier] Sections of the locspec.ctrl File**proxy.ini File**

The `proxy.ini` file identifies the `.pro` files that are used to specify configuration settings. With respect to adding location information to an `Access-Accept`, the `.pro` files are needed to invoke the `LOCSPEC` plugin.

Table 36 defines the fields needed in the `[Realms]` section for adding location-specific attributes to an `Access-Request`.

Table 36: proxy.ini [Realms] Fields for Configuration of Location-specific Attributes

Field	Description
realm_name	Lists all the realms that can be included in an <code>Access-Request</code> . For every <code>realm_name</code> , there must be an associated <code>realm.pro</code> file. For example, if the <code>[Realms]</code> section contains the lines: [Realms] CountryNet=countrynet.com There must be an associated <code>countrynet.pro</code> file.

Example

```
[Realms]
Realm_Example_1=nationwide.com
Realm_Example_2=peoplesnetwork.com
```

realm.pro File

The *realm.pro* file specifies the control point plug-in that is needed for attaching location-specific information to an Access-Request if the Access-Request is proxied from a Foreign AAA server to the Home AAA server.

Add the field `LOCSPEC` to both the `[Auth-Outbound-To-Proxy]` section and the `[Acct-Outbound-To-Proxy]` section in the *realm.pro* file. These sections call the location-specific control plug-in when an Access-Request is proxied (forwarded) to a Home AAA server.

Example realm.pro file:

```
[Auth-Outbound-To-Proxy]
LOCSPEC
[Acct-Outbound-To-Proxy]
LOCSPEC
⋮
```



NOTE: The `[Auth-Outbound-To-Proxy]` section and the `[Acct-Outbound-To-Proxy]` sections are the sections required in the *realm.pro* files that are related to adding location information to an Access-Request. However, the *realm.pro* files require additional sections that are related to the functionality of Steel-Belted Radius. See the *Steel-Belted Radius Administration Guide* and the *Steel-Belted Radius Reference Guide* for more information about proxy realm configuration.

Example Configuration for Adding NAS Location to Access-Request

Figure 14 on page 91 shows a sample configuration. The purpose of this example configuration is to add NAS location information to Access-Requests for `NAS_1`.

Example Overview

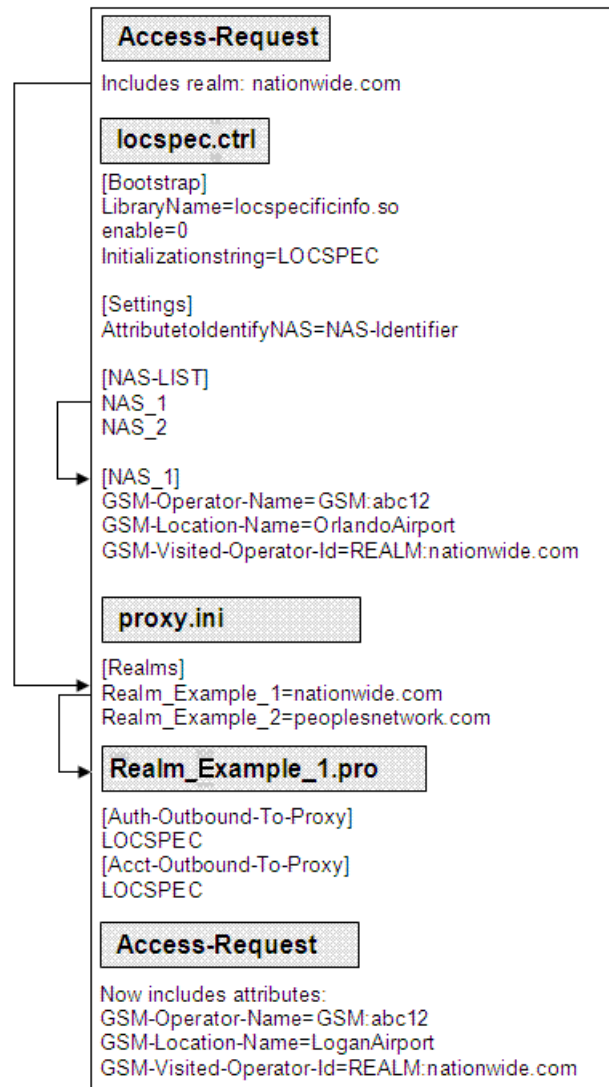
In this example, an Access-Request is sent for a mobile device through an example NAS identified by the name `NAS_1`. The example realm associated with the mobile device is `nationwide.com`. Three location attributes are assigned to `NAS_1` and included in the Access-Request that goes to the `nationwide.com` service provider. These three attributes are `GSM-Location-Name`, `GSM-Operator-Name`, and `GSM-Visited-Operator-Id`.

Example Configuration

The example configuration lines and syntax (shown in Figure 14 on page 91) associate all the configuration files together to attach NAS location information to an Access-Request.

The example configuration shows that if the realm is nationwide.com, then the .pro file to be used is Realm_Example_1.pro. The file Realm_Example_1.pro turns on NAS location information feature with the LOCSPEC commands.

Figure 14: Example Configuration for Adding Location Information to an Access-Accept



Assigning IP Addresses Based on Access Point Name (APN)

Steel-Belted Radius/SIM Server can assign IP addresses to mobile devices by access point (AP).

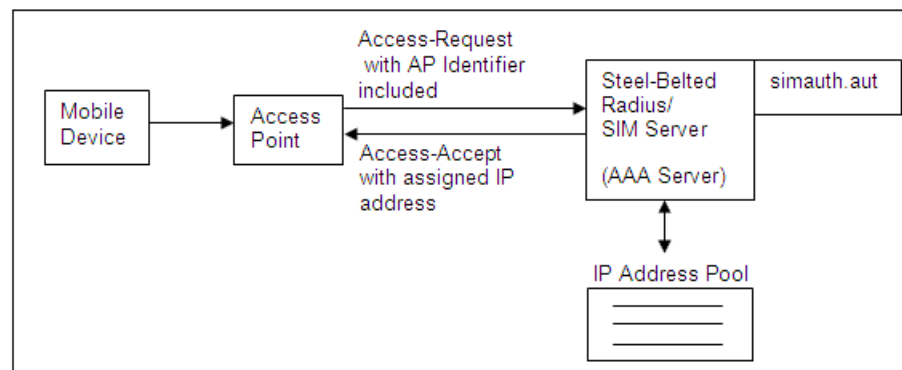
Overview

APN-based IP address assignment enables Steel-Belted Radius to perform the task of address assignment, rather than requiring the AP to assign addresses.

You can set up pools of IP addresses. Each pool consists of a set of IP addresses that can be assigned to a mobile device. You also configure an access point name (APN) to be associated with a particular pool. When an Access-Request is received, Steel-Belted Radius/SIM Server selects an IP address from the pool that is assigned to the APN that is handling the request.

Figure 15 shows the configuration of IP address assignment based on AP.

Figure 15: IP Address Assignment Based on Access Point



NOTE: APN-based IP address assignment takes precedence over all other methods of IP address assignment except if an IP address (or pool name) is added to an Access-Accept as the value of the Framed-IP-Address attribute.

For information about how to add any attribute from a subscriber database (such as a SQL database), see “Adding Attributes to an Access-Accept” on page 96. For example, you can retrieve the IP address from a SQL database and include it as the value of Framed-IP-Address in an Access-Accept

Tasks for Assigning IP Address Based on Access Point

Assigning IP addresses based on access point involves the following main tasks:

- Configure `simauth.aut`
- Create an address pool

Each of these tasks is described in the following sections.

Configuring `simauth.aut` for IP Address Assignment

The `simauth.aut` configuration file retrieves the IP address from a pool to be returned with the Access-Accept.

To configure `simauth.aut` for IP address assignment based on access point:

1. In the [Settings] section of `simauth.aut`, define the attribute that identifies the access point. This attribute is usually 3GPP-WLAN-APN-Id or Called-Station-ID. Use the following format:

```
AssignIpPoolByAttr = attribute
```

where:

attribute is the name of the string type attribute to be used to identify the Access Point.

Example:

```
[Settings]
AssignIpPoolByAttr = 3GPP-WLAN-APN-Id
```

2. In the [Settings] section of `simauth.aut`, define the attribute to contain the IP address to be assigned to the mobile device. This attribute will be returned with the Access-Accept. Use the following format:

```
AssignIpPoolDestAttr = attribute_for_address
```

where:

attribute_for_address is the attribute to be used to return the IP address in the Access-Accept. This attribute must be consistent with an IPv4 IP address. (It will usually be Framed-Ip-Address.)

Example:

```
[Settings]
AssignIpPoolDestAttr = Framed-Ip-Address
```

3. In the [IpPools] section of `simauth.aut`, define the Access Point Identifiers that associate specific pools with Access Points. Use the following format:

```
attribute-value = poolname
```

where:

attribute-value is the Access Point identifier.

poolname is the name of an IP address pool created using the Steel-Belted Radius Administrator. (See “Creating Address Pools” on page 95.)

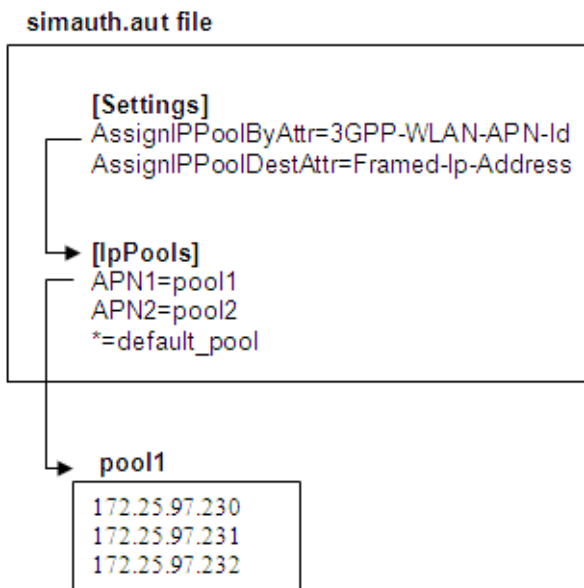
Example:

```
[IpPools]
ASN1 = Pool1
ASN2 = Pool2
* = Default_Pool
```

Figure 16 shows the configuration of `simauth.aut` for assigning IP address based on APN. In Figure 16, the AP is identified by the value of the attribute `3GPP-WLAN-APN-Id`. If the value of `3GPP-WLAN-APN-Id` is `APN1`, then an IP address will be taken from `pool1`. If the value is neither `APN1` nor `APN2`, the address will be taken from the pool named `default_pool`. The IP address is assigned to the attribute `Framed-IP-Address`. This attribute will be returned in the `Access-Accept`.

You create the IP address pools, `Pool1`, `Pool2`, and `Default_Pool`, using the Steel-Belted Radius Administrator as described in “Creating Address Pools” on page 95.

Figure 16: Configuration of IP Address Assignment Based on Access Point



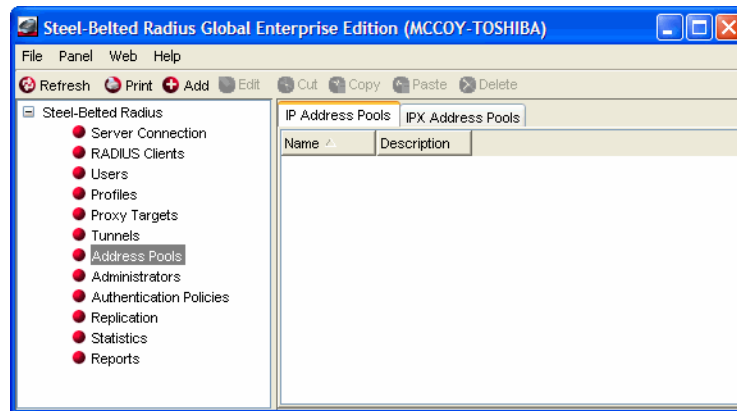
Creating Address Pools

Address pools are sets of IP addresses from which an IP address is assigned to a mobile device. You create address pools using the Steel-Belted Radius Administrator. See the *Steel-Belted Radius Administration Guide* for complete information about administering address pools.

To create address pools:

1. Click the **Address Pools** button to display the Address Pools panel (Figure 17).

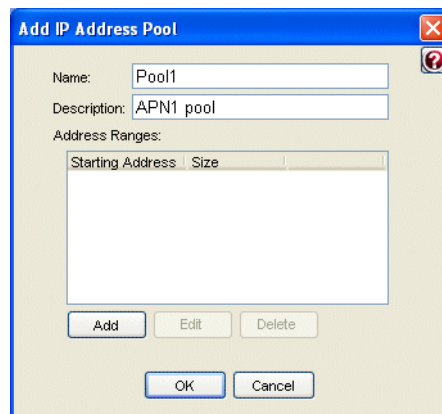
Figure 17: Address Pools Panel: IP Address Pools Tab



2. If necessary, click the **IP Address Pools** tab to display the list of IP address pools that have been configured.
3. Click the **Add** button.

The Add IP Address Pool dialog (Figure 18) appears.

Figure 18: Add IP Address Pool Dialog

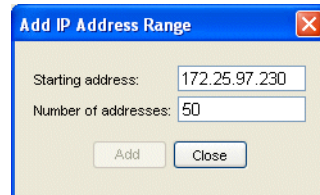


4. Enter the name of the IP address pool in the **Name** field.
5. Optionally, enter a description of the address pool in the **Description** field.
6. Identify the address ranges in the IP address pool.

- a. Click the **Add** button below the Address Ranges list.

The Add IP Address Range dialog (Figure 19) opens.

Figure 19: Add IP Address Range Dialog



- b. Enter the first address in the in the **Starting address** field.
 - c. Enter the number of addresses in the address range in the **Number of addresses** field.
 - d. Click **Add**.
 - e. Repeat steps b–d for each address range in the IP address pool. When you are finished, click **Close**.
7. Click **OK**.

Adding Attributes to an Access-Accept

You might want an Access-Accept to include attribute values. These attribute values can be retrieved from a subscriber database. For example, you might want to include the subscriber’s level of service in the Access-Accept as the value of the attribute Reply-Message. Another example might be retrieving the IP address to be assigned to a mobile node and returning it in the Access-Accept as the value of the attribute Framed-IP-Address.

Overview

An Access-Accept can include attribute values. Two authentication plugins are used to accomplish the tasks of authentication and adding attributes to an Access-Accept. The authentication plugins are:

- SIMAuth (acting as an “EAP helper”)

This authenticator provides EAP authentication.
- “Helped” Authenticator (usually SQLAuth)

This authenticator accesses the database, retrieves the specified attributes, and attaches them to the Access-Accept. In this situation, the “helped” authenticator does not perform any authentication tasks and its password-checking is suppressed. All authentication is performed by the SIMAuth, the EAP helper.

Data Flow

Authentication of the Access-Request and the addition of attributes to the Access-Accept is handled according to the following flow of data:

1. The mobile device sends an Access-Request to Steel-Belted Radius.
2. SIMAuth manages the EAP negotiation (challenge, and response).
3. If SIMAuth authenticates the request, it attaches the IMSI and MSISDN of the mobile device, and sends the request to SQLAuth.
4. SQLAuth can use the IMSI or MSISDN as a key to query the database and request attribute values (as a separate step from the SIMAuth authentication).
5. “Helped Authenticator” (usually SQLAuth) returns the Access-Accept with attribute values attached.

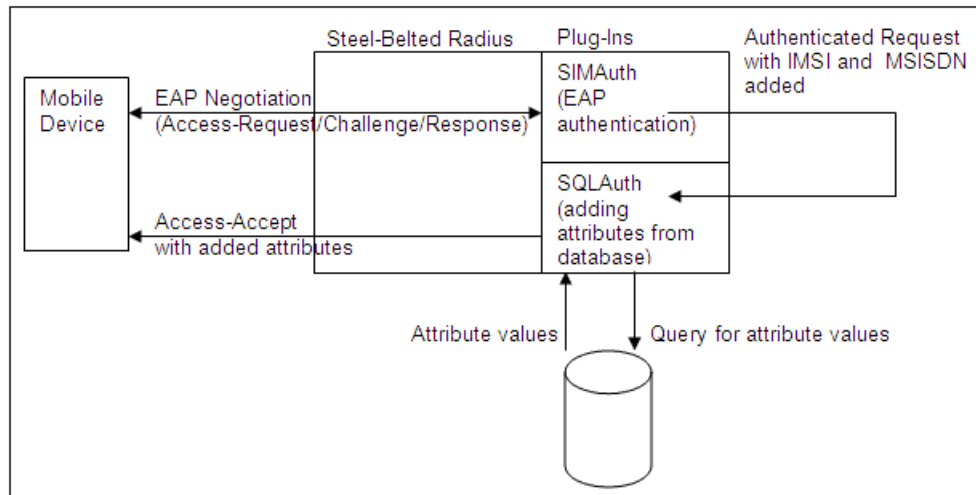


NOTE: SIMAuth is known as a Steel-Belted Radius “EAP helper” because it performs the EAP authentication for the “helped” authentication method (SQLAuth). Although SQLAuth is usually used for authentication, in this case the function of SQLAuth is to access the subscriber database, retrieve attributes, and return them with the Access-Accept.

For complete information about EAP helpers, see the *Steel-Belted Radius Administration Guide*.

Figure 20 shows an example data flow in which Steel-Belted Radius, SIMAuth, and SQLAuth work together to perform the following tasks:

- Access authentication (performed by SIMAuth)
- Addition of MSISDN and IMSI to the request (performed by SIMAuth)
- Database access and attribute retrieval (performed by SQLAuth in this example)
- Addition of retrieved attributes to the Access-Accept (performed by SQLAuth)

Figure 20: Example Data Flow for Addition of Attribute to Access-Accept

Configuration Tasks

To add attributes to the Access-Accept, you need to perform the following main tasks:

- Configure the related files, as described in “Configuring Files for Adding Attributes to Access-Accept” on page 98.
- Activate authentication as described in “Activating Authentication” on page 103.

Configuring Files for Adding Attributes to Access-Accept

The following files require special configuration to allow the addition of attributes to the Access-Accept:

- `simauth.aut`
- `simauth.eap`
- `radsql.aut`, `radsqljdbc.aut`, or `radldapauth.aut`
- `eap.ini`

To configure files for adding attributes to Access-Accept:

1. In the [Bootstrap] section of `simauth.aut` (for Oracle databases), set Enable to 0.

Setting Enable=0 ensures that these files are disabled.

Example:
 [Bootstrap]
 Enable=0

2. Create a copy of `simauth.aut` and name it `simauth.eap`.

This renaming causes SIMAuth to become the EAP helper.

3. In the [Bootstrap] section of `simauth.eap`, ensure that `Enable` is set to 1.
4. Open the relevant database access configuration file. This file is one of:
 - `radsq1.aut`
 - `radsq1jdbc.aut`
 - `radldapauth.aut`
5. Check the [Bootstrap] section of `radsq1.aut`, `radsq1jdbc.aut`, or `radldapauth.aut` for the name of the specified authentication method. In the following example, the specified authentication method is `SQLAuth`.

Example:
 [Bootstrap]
 Initializationstring=SQLAuth

For more information about how to configure the `radsq1.aut`, `radsq1jdbc.aut`, or `radldapauth.aut` files, see the *Steel-Belted Radius Administration Guide*.

6. Ensure that there is a section in the `eap.ini` file that has the name of the “helped” authentication method.

Example:
 [SQLAuth]

7. Ensure that the following lines to the “helped” authentication method section in `eap.ini` that you created in step 6.

```
[SQLAuth]
EAP-Only=1
First-Handle-Via-Auto-EAP=1
EAP-Type=SIM,AKA
Available-EAP-Only-Values=1
Available-Auto-EAP-Values=1
Available-EAP-Types=SIM|AKA
```



NOTE: The lines added in step 7 configure the specified authentication method (`SQLAuth`) and also prevent it from being used without the EAP helper (`SIMAuth`). The use of the “helped” authentication method (`SQLAuth`) without the EAP helper must be prevented because password checking is suppressed and the EAP helper (`SIMAuth`) is needed to perform authentication.

8. Suppress database password checking in the “helped” authentication method as described for Oracle, JDBC, and LDAP databases.
 - Oracle or JDBC: Do not provide a password in the `SQL=SELECT` statement in the [Settings] section of `radsql.aut` or `radsqljdbc.aut`. In the [Results] section of these files, include a `PASSWORD=` statement, leaving the password blank.

Example:
[Results]
Password=
 - LDAP: Remove the `%password=` setting from the [Response] section.
9. Insert a query into `radsql.aut`, `radsqljdbc.aut`, or `ldapauth.aut` to select the attributes to be added to the Access-Accept.

The selection of attributes from the database can be based on the database key values for IMSI or MSISDN. The values for IMSI or MSISDN are added to the request by SIMAuth in the attributes 3GPP-IMSI or Funk-SS7-MSISDN so that they can be used in the database query.

Example:
`SQL=SELECT subscriber-level FROM table 1 WHERE IMSI=@3GPP-IMSI`



NOTE: You can also use the `%username` or `%user` variables in the database query. However, they will not contain the expected values if pseudonyms are active.

10. Activate authentication as described in “Activating Authentication” on page 103.

Example Configuration for Adding Attributes to Access-Accept

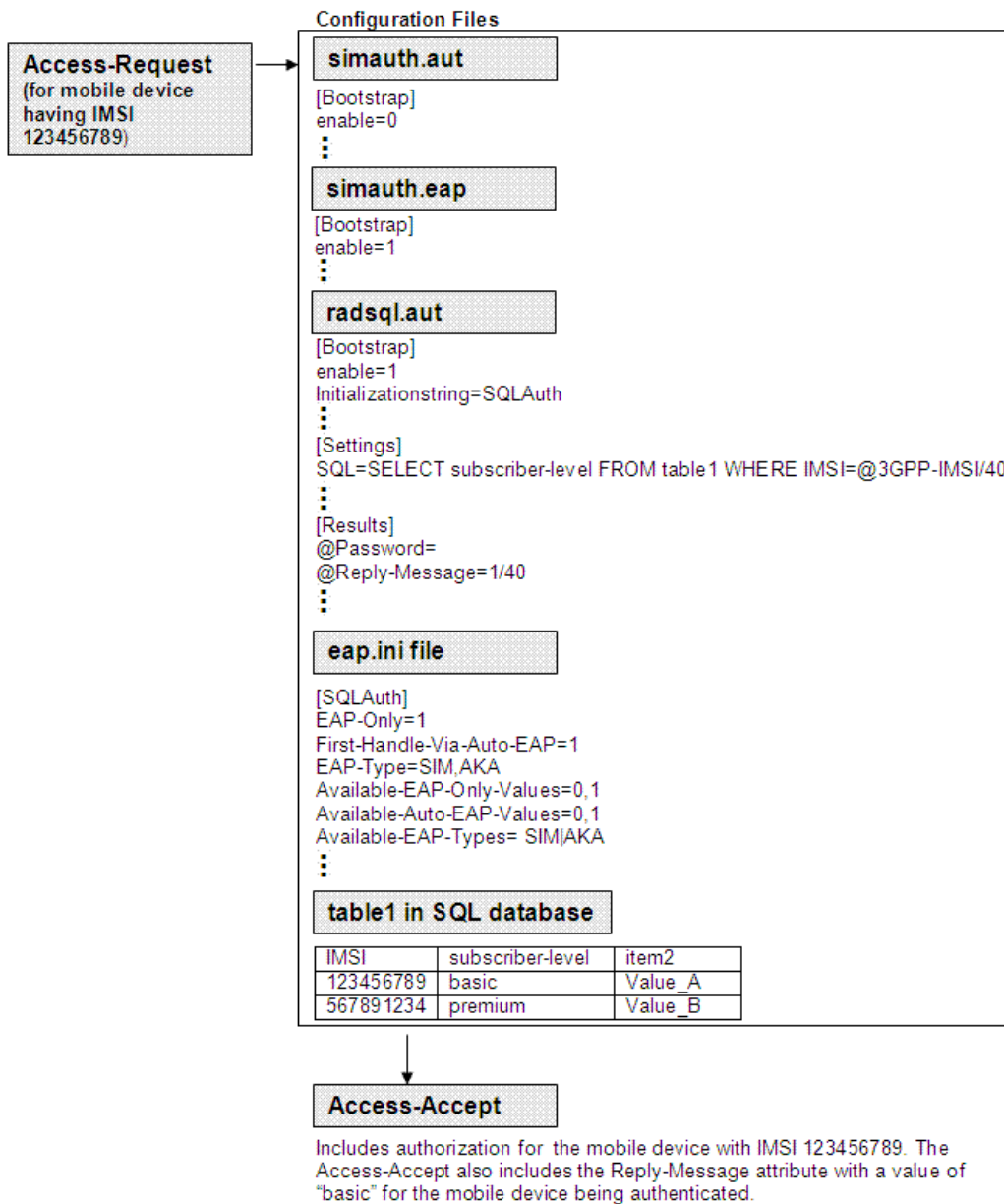
Figure 21 on page 101 shows a sample configuration. The purpose of this configuration is to query the database for a subscriber-level value and return the subscriber-level value along with the Access-Accept.

Example Overview

In this example, an Access-Request is sent for a mobile device with IMSI 123456789. The value of the subscriber-level for this device is retrieved from the database, assigned to the attribute Reply-Message, and attached to the Access-Accept.

The configuration lines and syntax (shown in Figure 14) associate all the configuration files together to attach an attribute to the Access-Accept.

Figure 21: Example Configuration for Adding Attributes to an Access-Accept



Example Notes

The sample configuration shown in Figure 14 on page 91 configures the data flow in the following way:

Access-Request

An Access-Request is sent to Steel-Belted Radius for the user with an IMSI value of 123456789.

SIMAuth

Simauth.eap file is enabled
Simauth.aut file is disabled.

Radsql.aut

[Bootstrap] section contains the name of the specified authentication method (SQLAuth). You later add a SQLAuth section to the eap.ini file.

Enter a SQL=SELECT statement to retrieve data from the database based on the value of the IMSI in the Access-Request. Do not include a password in the SQL SELECT statement.

The @Password= statement suppresses password checking of the database.

The "@Reply-Message=1/40" field indicates the following:

- The Reply-Message attribute will be added to the Access-Accept and carry the value retrieved from the database.
- The "1" in @Reply-Message=1/40 indicates that the first item in the SQL=SELECT statement (subscriber-level) is the column name of the SQL database from which the value is selected.
- The "40" in @Reply-Message=1/40 indicates that the width of the subscriber-level column is 40 characters.

Eap.ini

The eap.ini file must contain a section corresponding to the name of the "helped" authentication method named in the Initializationstring statement in the radsql.aut file. [SQLAuth] in this example.

The eap.ini file must contain the lines shown in Figure 14 on page 91 to configure SQLAuth and to prevent SQLAuth from acting without the SIMAuth.

Password-checking by SQLAuth is suppressed and the only authentication is being performed by SIMAuth (the EAP helper).

SQL database table 1

In this example, the SQL database is queried by SQLAuth and the subscriber-level for IMSI 123456789 is found to be "basic."

Access-Accept

The value of basic is assigned to the attribute Reply-Message and included in the Access-Accept.

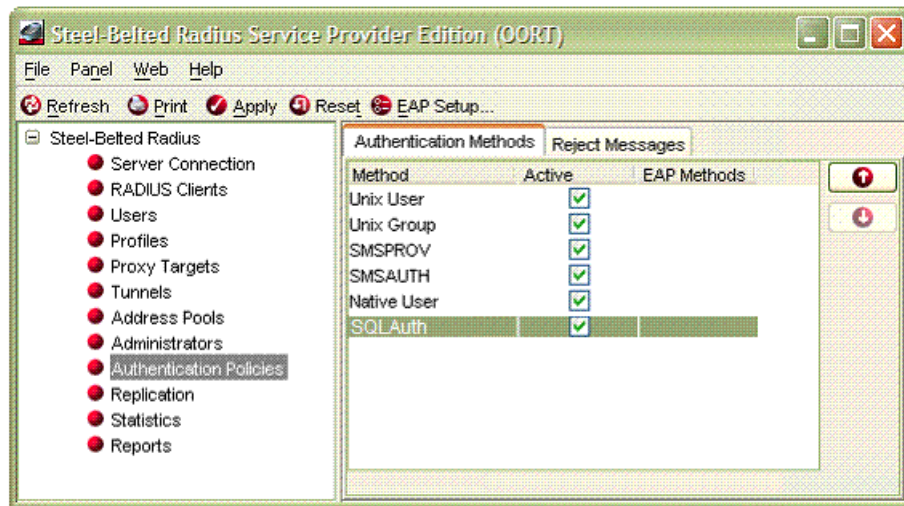
Activating Authentication

You activate authentication with the Steel-Belted Radius Administrator.

To activate authentication:

1. Run the Steel-Belted Radius Administrator and log into your Steel-Belted Radius server.
2. Click **Authentication Policies**.
3. Select the **Authentication Methods** tab.
4. Select the “helped” authentication method and select the check box to make it active.

Figure 22: SQLAuth, the “Helped” Authentication Method, in the List of Authentication Methods

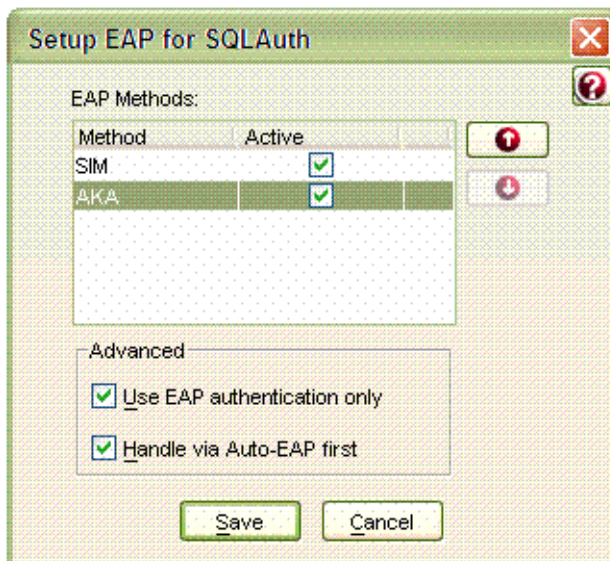


The name of the EAP helper (usually SIMAuth) does not appear in the list of Authentication Methods. For example, if SIMAuth was formerly used as an authentication method and is now authenticating as an EAP helper, SIMAuth no longer appears in the list of Authentication Methods. However, the name of the “helped” authentication method (usually SQLAuth) now appears in the list.

5. Double-click the name of the “helped” authentication method or click **EAP Setup**.

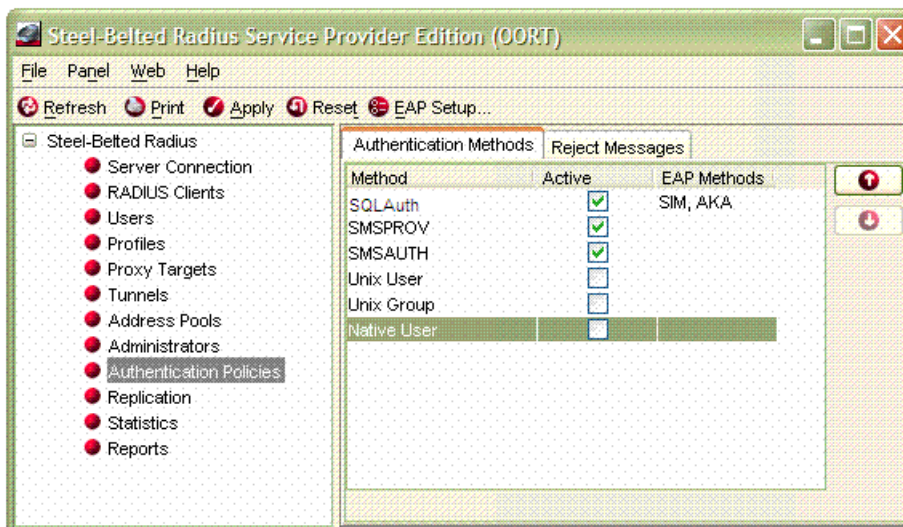
The Setup EAP Methods dialog appears.

Figure 23: Setup EAP Methods Dialog



6. Select all the check boxes so that SIM and AKA EAP methods are both enabled.
7. Click **Save**.
8. Optionally, change the order of the authentication methods so the “helped” authentication method is first. Disable authentication methods that are no longer applicable.

Figure 24: SQLAuth, the “Helped” Authentication Method, at the Top of the List of Authentication Methods



Chapter 6

Configuring the Call Detail Record Accounting Module

Steel-Belted Radius/SIM Server includes a Call Detail Record (CDR) accounting module. The CDR module manages all CDR-based subscriber accounting for the purposes of billing. CDRs can be generated for authentications performed by `simauth`, `ldapauth`, or `sqlauth`.

Overview of CDR Process

The following steps take place after account provisioning and authentication:

1. After it receives a RADIUS Access-Accept message, the Access Controller grants the subscriber access to the network, and sends a RADIUS Accounting-Start message to the Steel-Belted Radius/SIM Server.
2. Steel-Belted Radius/SIM Server passes the Accounting-Start message to the Call Detail Record (CDR) accounting module (`cdRACT`) and determines if accounting requests have been previously received for the account.
3. Steel-Belted Radius records fixed-fee, session, or partial CDRs to the accounting file system.
4. The service provider periodically exports CDR billing information to a billing application.

Types of Call Detail Records

Steel-Belted Radius/SIM Server can forward call detail records to a billing gateway when subscribers initiate and terminate sessions. Steel-Belted Radius/SIM Server can produce three types of CDR:

- **Fixed-fee CDRs**—Generated when a subscriber logs into the SMS network and receives a password in the form of a text message. A fixed-fee CDR provides network access for a specified time period (regardless of the volume of data sent or received over the WLAN link). Fixed-fee CDRs can be generated during SMS authentication. Fixed-fee CDRs are not generated during SIM authentication.

- **Session CDRs**—Generated at the end of each WLAN session. A session CDR records the length of time of the WLAN session and the number of bytes of data sent and received. Session CDRs can be generated during SMS authentication or SIM authentication.
- **Partial CDRs**—Generated periodically during a WLAN session either (a) after a specified amount of time has elapsed or (b) after a session has transferred a specified threshold quantity of data. A provider can specify how frequently a system generates a partial CDR by specifying time and volume (number of bytes) settings in the `cdracct.acc` file. After a partial CDR is generated for a subscriber session because a time or volume threshold is exceeded, both threshold triggers are reset to zero.

Partial CDRs can be generated during SMS or SIM authentication. A provider can configure time and volume thresholds for SIM authentication and SMS authentication separately.

Configuring Accounting Options with `cdracct.acc`

You can configure the available options for CDR accounting in the `cdracct.acc` [Settings] section.

[Bootstrap] Section of `cdracct.acc`

The [BootStrap] section of the `cdracct.acc` file contains the settings shown in Table 37.

Table 37: `cdracct.acc` [Bootstrap] Fields

Field	Description
LibraryName	The name of the library called when <code>cdracct</code> runs. The default value is <code>cdracct.so</code> .
Enable	Set to 1 to enable the features described in this file. Set to 0 to disable the features described in this file. Default is 1.
InitializationString	The <code>cdracct</code> initialization string. Default is <code>CDRACCT</code> .

Example

```
[Bootstrap]
LibraryName=cdracct.so
Enable=1
InitializationString=CDRACCT
```

[Settings] Section of cdracct.acc

The [Settings] section of the cdracct.acc file contains the settings shown in Table 38.

Table 38: cdracct.acc [Settings] Fields

[Settings] Field	Description
ConfigLog	<p>Specifies the method for logging cdracct.acc configuration information:</p> <p>None — Configuration information will not be captured.</p> <p>ConsoleAndLog — Sends the log information to both the console and the log.</p> <p>Console — Sends the log information to the console only.</p> <p>Log — Sends the log information to the log only.</p>
CDRDirectory	<p>The directory where the CDR records are stored.</p> <p>The default value is <code>./CDR</code> under the Steel-Belted Radius install directory. If you modify this directory name, make sure it exists before you use Steel-Belted Radius/SIM Server.</p>
CDRNodeID	<p>The name for the authentication server machine in the NodeID field of the generated CDR.</p> <p>If CDRNodeID is omitted or commented out, the correct value is updated automatically by Steel-Belted Radius/SIM Server on system startup. This value is the Solaris system machine name for the machine on which Steel-Belted Radius/SIM Server is running.</p>
UserPartialCdrEnable	<p>Specifies whether or not partial CDRs are generated for LDAPauth or SQLauth users.</p> <ul style="list-style-type: none"> ■ 0 — Partial CDRs are not generated. ■ 1 — Partial CDRs are generated. <p>The default is 0.</p> <p>If UserPartialCdrEnable = 1, then UserSessionCdrEnable must be set to 0.</p>
UserSessionCdrEnable	<p>Specifies whether or not session CDRs are generated for LDAPauth or SQLauth users.</p> <ul style="list-style-type: none"> ■ 0 — Session CDRs are not generated. ■ 1 — Session CDRs are generated. <p>The default is 1.</p> <p>If UserSessionCdrEnable = 1, then UserPartialCdrEnable must be set to 0.</p>
SMSFixedFeeCdrEnable	<p>Specifies whether or not fixed-fee CDRs are generated for SMS users.</p> <ul style="list-style-type: none"> ■ 0 — Fixed-fee CDRs are not generated. ■ 1 — Fixed-fee CDRs are generated. <p>The default is 1.</p>

Table 38: cdracct.acc [Settings] Fields (continued)

[Settings] Field	Description
SMSPartialCdrEnable	<p>Specifies whether or not partial CDRs are generated for SMS users.</p> <ul style="list-style-type: none"> ■ 0 – Partial CDRs are not generated. ■ 1 – Partial CDRs are generated. <p>The default is 0.</p> <p>Note: If SMSPartialCdrEnable = 1, then SMSSessionCdrEnable must be set to 0.</p>
SMSSessionCdrEnable	<p>Specifies whether or not session CDRs are generated for SMS users.</p> <ul style="list-style-type: none"> ■ 0 – Session CDRs are not generated. ■ 1 – Session CDRs are generated. <p>The default is 1.</p> <p>Note: If SMSSessionCdrEnable = 1, then SMSPartialCdrEnable must be set to 0.</p>
SIMPatialCdrEnable	<p>Specifies whether or not partial CDRs are generated for SIMAuth users.</p> <ul style="list-style-type: none"> ■ 0 – Partial CDRs are not generated. ■ 1 – Partial CDRs are generated. <p>The default is 0.</p> <p>Note: If SimPartialCdrEnable = 1, then SimSessionCdrEnable must be set to 0.</p>
SIMSessionCdrEnable	<p>Specifies whether or not session CDRs are generated for SIMAuth users.</p> <ul style="list-style-type: none"> ■ 0 – Session CDRs are not generated. ■ 1 – Session CDRs are generated. <p>The default is 1.</p> <p>NOTE: If SimSessionCdrEnable = 1, then SimPartialCdrEnable must be set to 0.</p>
VolumeThresholdMegaBytes	<p>Specifies the threshold (in megabytes) for creating a partial CDR.</p> <p>Default is 10 megabytes.</p>
TimeThresholdSeconds	<p>Specifies the threshold (in seconds) for creating a partial CDR.</p> <p>Default is 600 seconds (10 minutes).</p> <p>Note: The value entered for TimeThresholdSeconds must match the value specified for the Acct-Interim-Interval return list attribute for the user (or the profile assigned to the user) in Steel-Belted Radius.</p>
VolumeThresholdEnable	<p>Specifies whether partial CDRs are generated when the volume threshold is crossed.</p> <ul style="list-style-type: none"> ■ 0 – Partial CDRs are not generated. ■ 1 – Partial CDRs are generated. <p>The default is 0.</p>

Table 38: cdracct.acc [Settings] Fields (continued)

[Settings] Field	Description
TimeThresholdEnable	<p>Specifies whether partial CDRs are generated when the time threshold is crossed.</p> <ul style="list-style-type: none"> ■ 0 – Partial CDRs are not generated. ■ 1 – Partial CDRs are generated. <p>The default is 0.</p>
CdrDownlink	<p>Specifies the RADIUS attribute to which the CDR Downlink field is mapped. Options are:</p> <ul style="list-style-type: none"> ■ Acct-Input-Octets ■ Acct-Output-Octets <p>The default is Acct-Input-Octets.</p> <p>Note: The CDR Uplink field is automatically mapped to whichever attribute is not assigned to the Downlink field.</p>
CdrType	<p>Specifies the format for the CDR type.</p> <ul style="list-style-type: none"> ■ BinaryV1 – Version 1 type CDRs are generated with extension .cdr1. ■ BinaryV2 – Version 2 type CDRs are generated with extension .cdr2. ■ Asn1V2 – ASN.1 type CDRs are generated with extension .cdr2a. <p>If CdrType is not specified, Version 1 type CDRs are generated with extension .cdr. Files with both .cdr and .cdr1 extensions are identical (version 1). The .cdr extension is retained for backward compatibility.</p>
DefaultCUIDType	<p>Specifies the user id to be used when no CUID (ChargeableUserId) attribute is received with the accounting request. This setting applies only if CdrType is set to BinaryV2 or Asn1V2.</p> <p>Allowed values are:</p> <ul style="list-style-type: none"> ■ IMSI ■ MSISDN ■ NAI <p>The default value is NAI.</p>

Example

```
[Settings]
ConfigLog=ConsoleAndLog
CDRDirectory=./CDR
CDRNodeID=
SMSFixedFeeCdrEnable=1
SMSPartialCdrEnable=0
SMSSessionCdrEnable=1
SMPartialCdrEnable=0
SIMSessionCdrEnable=1
UserPartialCdrEnable=0
UserSessionCdrEnable=1
VolumeThresholdMegaBytes=10
TimeThresholdSeconds=600
VolumeThresholdEnable=0
```

```

TimeThresholdEnable=0
MinimumTimeThresholdSeconds=60
CdrDownlink=Acct-Input-Octets
CdrType=BinaryV2
DefaultCUIDType=NAI
.

```



NOTE: The radius.ini file installed with Steel-Belted Radius must contain the following lines to ensure that the MSISDN is transported to the cdracct.act plug-in.

```

[EmbedInClass]
Funk-SS7-MSISDN=Encrypt,Remove

```

Displaying CDR Information

CDRs are created periodically (for partial CDRs), at the end of a session (for session CDRs), or at logon (for fixed fee). You set fields in the [Settings] section of cdracct.act to generate CDRs within sessions. See Table 38 on page 107 for a list of settings that affect CDR creation.

CDR Files

CDR filenames are assigned to CDRs with the filename incrementing by one for every CDR file generated within the same sessions. All CDR files have a .cdr, .cdr1, .cdr2, or .cdr2a extension, depending on the setting for the CdrType field in the cdracct.act file described in Table 38 on page 107.

Table 39: CdrType Settings in cdracct.act and Resulting CDR Versions and Filename Extensions

CdrType setting in cdracct.act file	CDR Version Generated	CDR filename extension
CdrType = BinaryV1	Binary Version 1	.cdr1
CdrType = BinaryV2	Binary Version 2	.cdr2
CdrType = Asn1V2	ASN1 Version 2	.cdr2a
no setting specified	Binary Version 1	.cdr

The following example shows a listing of the CDR files in the CDR directory.

```

$ ls
4436830d_00000003.cdr1 44368607_00000003.cdr1 4436865f_00000003.cdr1
4436830d_00000004.cdr2 44368607_00000004.cdr2 4436865f_00000004.cdr2
4436830d_00000001.cdr2a 44368607_00000001.cdr2a 4436865f_00000001.cdr
443c0b94_00000001.cdr 4436830d_00000002.cdr2 44368607_00000002.cdr2
4436865f_00000002.cdr2

```

Using *cdrdump* to Display CDR File Contents

CDR files are binary or ASN.1 type files. The *cdrdump* tool provided with SIM Server displays the contents of a CDR file.



NOTE: Use the *CdrType* field in the [Settings] section of the *cdracct.acc* file to set the type of CDR file (binary version 1, binary version 2, or ASN.1). See Table 38 on page 107 for more information.

To display CDR contents using *cdrdump*:

1. Go to the *radius/CDR* directory and enter the list command to view the CDR filenames. For example:

```
$ ls
```

2. Enter the *cdrdump* command in the following format:

```
$ ../cdrdump [-r] filename
```

where:

filename is the CDR filename as described in “CDR Files” on page 110.

-r indicates that the display should be in raw format.

The *-r* raw format command produces unformatted data but does display field names and datatypes. In raw format, every byte is displayed, even the insignificant bytes (such as those past the end-of-string NUL character). Omitting the *-r* field causes the display to appear in formatted mode. Table 40 on page 113 lists the differences between raw mode and formatted mode.

You cannot use the *-r* field for ASN1 (extension *cdr2a*) files. To display ASN1 files in raw format, see “Displaying ASN1 CDR Files in Raw Format Using *dumpsasn1*” on page 113.



NOTE: You can send the output of *cdrdump* to a file with the command

```
$ ../cdrdump filename > output_filename.
```

- View the `cdrdump` output. Refer to “CDR Fields” on page 113 for information about each field of information. The following listing shows example output from a `cdr` file.

```

$ ./cdrdump 45081ce7_00000002.cdr2
DUMP OF CDRv2 FILE "45081ce7_00000002.cdr2":
RecordType..... : 95
ServedImsi..... : 212864080212345
ChargingId..... : 902
GgsnAddress..... : 0.0.0.0
NasAddress..... : 172.25.97.133
NasPortType..... : 0
NasTimeZone(15min,Dst)..... : 255,255
AsAddress..... : 172.25.98.242
NodId..... : sbrha-4
AccessPointName/NasId..... : 172.25.97.133
ProtocolType(Org,Val)..... : 1,33
MtAddress..... : 10.10.10.10
DataVolumeUplink..... : 10000000
DataVolumeDownlink..... : 2000000
RecordOpeningTime..... : Wed Sep 13 15:04:34 2006 UTC
ChangeTime..... : Wed Sep 13 15:04:54 2006 UTC
Duration..... : 20
CauseForRecordClosing..... : 0
RecordSequenceNumber..... : 1
ChargingType..... : 8
ChargingCharacteristics..... : 0
ConnectionType..... : 255
ServedMsisdn..... : 1212812345
ChargeableUidType..... : 1
ChargeableUidLength..... : 15
DomainIndex..... : 0
ChargeableUid..... : 212864080212345
LocationName..... : Airport
LocationInfo..... : country=US;A1=MA;A3=Boston;ZIP=02128
VisitedOperatorId..... : REALM:foobar.com
OperatorName..... : REALM:sim.com
ExtChargingIdLength..... : 2
ExtChargingId..... : 20
PdpChargingCharacteristics..... : 0
QosTrafficClass..... : 5
PdpAddress..... : 10.10.10.10

```



NOTE: You can use the UNIX `od` command to display the `cdrdump` file contents in purely raw format (all hexadecimal with no field names or datatypes displayed).

Example: `$- od -x frame`

cdrdump Output

You can specify that cdrdump output be formatted or raw using the `-r` switch with the `cdrdump` command. (See Step 2 in “Using cdrdump to Display CDR File Contents” on page 111.) Table 40 lists the differences between formatted and raw output.

Table 40: Differences Between Raw and Formatted cdrdump Output

Output Type	Formatted	Raw
Numbers	Decimal	Hexadecimal
IP addresses	IPV4 (dotted quad) if the first 12 bytes are zeroes. IPV6 if the first 12 bytes are non-zero.	IPV6
Timestamps	Version 1: Local time Version 2 and ASN.1: Universal Coordinated Time	Hexadecimal number of seconds since the UNIX epoch followed by ISO format
strings	Terminated at the first null character	Every character is included
BCD strings	Terminated at the first nibble inconsistent with BCD encoding (such as 0xf)	All bytes are displayed in hexadecimal
Version 1 reserved fields	Not displayed	Displayed
Bytes displayed	Only relevant characters are displayed.	Every byte is displayed, including those past the end-of-string null characters.

Displaying ASN1 CDR Files in Raw Format Using `dumpasn1`

To display ASN.1 CDRs in raw format, use the `dumpasn1` tool located in the `radius` directory. When invoking the `dumpasn1` command, always use the `-z` option to ensure that zero-length fields are displayed properly, as shown in the following example:

```
dumpasn1 -z filename.cdr2a
```



NOTE: CDR files for ASN.1 type can be displayed with formatting (not raw) using the `cdrdump` file as described in “Using cdrdump to Display CDR File Contents” on page 111.

CDR Fields

CDR fields can be of type Version 1, Version 2, or ASN.1. Table 41 describes the fields that are contained in the CDR and displayed using `cdrdump`. Table 42 on page 121 describes the field formats for Version 1 and Version 2 CDRs. Figure 25 on page 122 describes the field formats for ASN.1 CDRs.



NOTE: Use the `CdrType` field in the [Settings] section of the `cdracct.acc` file to set the type of CDR file (binary version 1, binary version 2, or ASN.1). See Table 38 on page 107 for more information.

Table 41: CDR Fields

Field Name in CDR Display	Present in Version 1	Present in Version 2 and ASN.1	Field Format	Meaning of Field in cdrdump
RecordType	Yes	Yes	V1/V2: Byte length 1 ASN.1: Integer	Access context of the record. It is always a "Wireless LAN context" record W-CDR. The value of this field is always 95.
ServedImsi	Yes	Yes	V1/V2: BCD encoding. Length 8 bytes. ASN.1: NumericString	IMSI of the served party, the user. If IMSI is not present or available, this field contains exactly the home operator's MCC + MNC.
ChargingId	Yes	Yes	Integer length 4	Particular session of the user, together with the GGSN (or AC) address and the Record Sequence Number. Subsequent sessions of a user have a different Charging ID.
GgsnAddress	No	Yes	V2: IP-Address length 16 ASN.1: IPV6 Address	Specifies IP address of the GGSN in IPv6 format; for IPv4 addresses, the first 12 bytes are all 0x0. If the user is not connected with GGSN, then the IP address should be ACs IP address.
NasAddress	Yes	Yes	V1/V2: IP-Address length 16 ASN.1: IPV6 Address	IP address of the Network Access Server (Access Point in 802.1x and Access Controller in Open System) used for the session. Field is in IPv6 format; for IPv4 addresses, the first 12 bytes are all 0x00.
NasPortType	No	Yes	V2: Integer length 4 ASN.1: Integer	15 = Ethernet, 19 = 802.11. Defines the port type. Value directly from GSMA Vendor-Specific "NAS port type" attribute received from NAS.
NasTimeZone	No	Yes	V2: Byte length 2 ASN.1: NAS-TimeZone	Field specifies the time zone and daylight saving usage. First byte indicates Time zone in 15 minutes intervals preceded by + for positive or - for negative from GMT. 2nd byte is daylight saving indication. 1 indicates 1 hour adjustment for Daylight Saving Time. 2 indicates 2 hour adjustment for Daylight Saving Time.
AsAddress	Yes	Yes	V1/V2: IP-Address length 16 ASN.1: IPV6 Address	Address of Application Server (AS) that generates the CDR in IPv6 format; for IPv4 addresses, the first 12 bytes are all 0x00.
Nodeld	Yes	Yes	V1/V2: Text length 20 ASN.1: UTF8String	Distinguished Name of the AS that created the record.

Table 41: CDR Fields (continued)

Field Name in CDR Display	Present in Version 1	Present in Version 2 and ASN.1	Field Format	Meaning of Field in cdrdump
AccessPoint Name/NasId	Yes	Yes	V1/V2: Text length 63 ASN.1: UTF8String	NAS-identifier or other available data concerning the location of access zone. This field can be zero.
ProtocolType	Yes	Yes	V1/V2: Word length 2 ASN.1: Protocol-Type	First byte is the "PDP type organization" (0 = ETSI, 1 = IETF). Second byte is the "PDP type value"; for ETSI, valid values are 0 (X.25), 1 (PPP) and 2 (OSP:IHOSS). For IETF valid values are HEX(21) (End User Address information element for IPv4) and HEX(57) (for IPv6). In OWLAN context, the PDP type organization is always 1 = IETF.
MtAddress	Yes	Yes	V1/V2: IP-Address length 16 ASN.1: IPV6 Address	IP address of the Mobile Terminal. IP address of the end-user's terminal can be sent in the Framed-IP-Address attribute in Accounting-Requests [RFC2866]. If the Framed-IP-Address attribute is present in Accounting-Request, AS will include that IP address in the MT Address field of CDR.
DataVolume Uplink	Yes	Yes	Integer length 4	Number of bytes transmitted from the MT since the opening of the CDR.
DataVolume Downlink	Yes	Yes	Integer length 4	Number of bytes transmitted towards the MT since the opening of the CDR
Record Opening Time	Yes	Yes	V1: Local time V2 and ASN.1: UTC	Number of times when the record was opened, that is, when an Accounting-Request with Acct-Status-Type Start (session started) or Interim-Update (partial CDR written) was received. In CDR, the time is AS local time.
ChangeTime	Yes	Yes	V1: Local time V2 and ASN.1: UTC	Time when the container was closed, that is, when an Accounting-Request with Acct-Status-Type Stop or Interim-Update was received. V1: Local time. V2 and ASN.1:Timestamp
Duration	Yes	Yes	Integer length 4	Duration of the session in seconds. This value is received from the NAS and is not necessarily the difference between Change Time and Record Opening Time. This field should be used as the basis for time-based billing.

Table 41: CDR Fields (continued)

Field Name in CDR Display	Present in Version 1	Present in Version 2 and ASN.1	Field Format	Meaning of Field in cdrdump
CauseFor RecordClosing	Yes	Yes	V1/V2: Byte length 1 ASN.1: Integer	Reason why the CDR is closed. The valid values for this field are: 0 indicates user logged out, lost service, NAS request, callback, or host request. 1 indicates partial CDR volume threshold exceeded. 2 indicates partial CDR time threshold exceeded. 7 indicates user's session was lost, session timeout, port error, NAS error, NAS reboot, port unneeded, port pre-empted, port suspended, service unavailable, user error, or idle timeout. 16 indicates Acct-Input/Output-Gigawords counter value has incremented. User has transferred over 2 ³² bytes. 20 indicates a management action caused session termination.
Record Sequence Number	Yes	Yes	Integer length 4	Running sequence number starting from 1, which is used to link charging records generated for a given end-user's session. Value is incremented for each partial record.
ChargingType	Yes	Yes	V1/V2: Byte length 1 ASN.1: Integer	Charging type is always 8 (normal postpaid record).
Charging Characteristics	Yes	Yes	V1/V2: Byte length 1 ASN.1: String	Type of CDR. CDRs are created based on the authentication method. The possible values are 0-EAP-SIM, 1-SMS Fixed fee CDR, 2-SMS Session CDR. These values are also used in partial CDRs.
Connection Type	No	Yes	Integer	Identifies the type of connection. 0 indicates direct 7 indicates GGSN
ServedMsIsdn	Yes	Yes	V1/V2: BCD encoding. Length 9 bytes. ASN.1: NumericString	Mobile Station ISDN number of the served party.

Table 41: CDR Fields (continued)

Field Name in CDR Display	Present in Version 1	Present in Version 2 and ASN.1	Field Format	Meaning of Field in cdrdump
ChargeableUid Type	No	Yes	V1/V2: Byte length 1 ASN.1: Integer	Field specifies the used type of ChargeableUid (Charging Type identifier). Charging Type identifiers are initially assigned as follows: 00 - reserved 01 - IMSI (example: 1231231231244...) 02 - NAI (example: foo@bar.com) 03 - E.164 (a MSISDN - example: +358405627015) 04 - TMPID (as described in 3GPP TS33.234 Temporary Identity Generation example) Note: TMPID not supported The choice of the identifier (IMSI, NAI, or MSISDN) is determined by the value set for ChargeableUserIdInResponse in the <code>simauth.aut</code> file. If more than one value is set, the identifier returned by the NAS is used. Usually, the identifier returned by the NAS is the first in the list of multiple identifiers specified for ChargeableUserIdInResponse in the <code>simauth.aut</code> file. For more information, see ChargeableUserIdInResponse in Table 44 on page 124. Also see ChargeableUidLength and ChargeableUid in this table.
ChargeableUid Length	No	Yes	V1/V2: Byte length 1 ASN.1: String	Specifies the length the “string” in Chargeable-User-ID. Valid only if the Charging Type Identifier value is 2. Also see ChargeableUidType and ChargeableUid in this table.
DomainIndex	Yes	Yes	V1/V2: Byte length 1 ASN.1: Integer	Location where the domain part starts in the Username and domain field. V2: Valid only if the Charging Type Identifier value is 2 (NAI).
UserNameAnd Domain	Yes	No	Text length 253	Username and realm/domain of the user in NAI format.

Table 41: CDR Fields (continued)

Field Name in CDR Display	Present in Version 1	Present in Version 2 and ASN.1	Field Format	Meaning of Field in cdrdump
ChargeableUid	No	Yes	V1/V2: Integer length 1 ASN.1: UTF8String	<p>Replaces User Name field in Version 1.</p> <p>GSMA-specified chargeable user with ChargeableUid Type and with ChargeableUid Length. The content string interpretation is based on the ChargeableUidType. (It is either the value of the IMSI, NAI, or MSISDN.)</p> <p>The choice of the identifier (IMSI, NAI, or MSISDN) is determined by the value set for ChargeableUserIdInResponse in the <code>simauth.aut</code> file. If more than one value is set, the identifier returned by the NAS is used. Usually, the identifier returned by the NAS is the first in the list of multiple identifiers specified for ChargeableUserIdInResponse in the <code>simauth.aut</code> file.</p> <p>If the NAS fails to return a CUID, the value set for <code>DefaultCUIDType</code> in the <code>cdracct.acc</code> file is used.</p> <p>For more information, see ChargeableUserIdInResponse in Table 44 on page 124.</p> <p>Also see ChargeableUidLength and ChargeableUidType in this table.</p>
LocationName	Yes	Yes	V1/V2: Text length 32 ASN.1: String	<p>Textual description of the WLAN Hot Spot. For example, "London City Airport." Human readable string without mandated format - printable. Attribute can be used:</p> <ul style="list-style-type: none"> ■ For a string information printed into subscriber's detailed bill ■ For bilaterally agreed data between operators <p>The contents are copied directly from GSMA Vendor-Specific Location-Name attribute received from NAS. Possible truncation may be done at the end of the string.</p> <p>V1: Format is WISPr. Source is WISPr specific attribute Location-Name.</p> <p>V2 and ASN.1:Format changed from WISPr to IR.61. Source is Vendor Specific Location-Name.</p>

Table 41: CDR Fields (continued)

Field Name in CDR Display	Present in Version 1	Present in Version 2 and ASN.1	Field Format	Meaning of Field in cdrdump
LocationInfo	Yes	Yes	V1/V2: Text length 64 ASN.1: String	<p>Location-Information attribute:</p> <ul style="list-style-type: none"> ■ The ISO 3166 country code is mandatory. ■ The location should identify the network (“what” is code 3). ■ Other recommended information includes: <ul style="list-style-type: none"> ■ A1 - state, region, province, or prefecture ■ A2 - county, parish, gun, or district ■ A3 - city or township ■ NAM - Name (residence, business or office occupant) ■ Additional location information fields are up according to bilateral agreements between operators. The contents are copied directly from GSMA Vendor-Specific Location-Info attribute received from NAS. Possible truncation may be done at the end of the string. <p>V1: Format is WISPr. Source is Vendor-Specific attribute “WISPr Location-ID.”</p> <p>V2 and ASN.1:Format changed from WISPr to IR.61. source is Vendor Specific Location-Info.</p>
Visited OperatorID	Yes	Yes	V1/V2: Text length 8 ASN.1: String	<p>Formatted ASCII string that has two parts separated with a colon.</p> <ul style="list-style-type: none"> ■ GSM:TADIG - Prefix string is “GSM” and the following code is a GSMA assigned TADIG code presented in capital ASCII letters. ■ REALM:realm - Prefix string is “REALM” and the following code is any valid domain name string that has been acquired from any valid registrar or registry. <p>The contents are copied directly from GSMA Vendor-Specific Visited-Operator-ID attribute received from NAS. Possible truncation may be done at the end of the string.</p> <p>V1: Format WISPr. Set to zero.</p> <p>V2 and ASN.1:Format changed from WISPr to IR.61. Source is Vendor Specific Visitor-Operator-ID.</p>

Table 41: CDR Fields (continued)

Field Name in CDR Display	Present in Version 1	Present in Version 2 and ASN.1	Field Format	Meaning of Field in cdrdump
OperatorName	No	Yes	V1/V2: Text length 128 ASN.1: String	<p>Contains a formatted ASCII string that has two parts separated with a colon:</p> <ul style="list-style-type: none"> ■ GSM:TADIG - Prefix string is “GSM” and the following code is a GSMA assigned TADIG code presented in capital ASCII letters. ■ REALM:realm - Prefix string is “REALM” and the following code is any valid domain name string that has been acquired from any valid registrar or registry. <p>The contents are copied directly from GSMA Vendor-Specific Visited-Operator-ID attribute received from NAS. Truncation can be done at the end of the string. The contents are copied directly from GSMA Vendor-Specific Operator-Name attribute received from NAS. Possible truncation can be done at the end of the string.</p> <p>If both Visited-Operator-ID and Operator-ID are present in the Access-Response, the Visited-Operator-ID is used.</p>
ExChargingId Length	No	Yes	V1/V2: Text length 1 ASN.1: Integer	Length of ExternalChargingID attribute. Contains length of ExChargingID. For example, if the value of the ExternalChargingID attribute is 20, the length is 2.
ExChargingId	No	Yes	V1/V2: Text length 1 ASN.1: String	Value of the RADIUS attribute, Acct-Session-ID.
PDPCharging Characteristics	No	Yes	V1/V2: Byte length 1 ASN.1: Integer	Charging type applied to PDP context.
QosTraffic Class	No	Yes	V1/V2: Byte length 1 ASN.1: Integer	Quality of Service. Possible values = 0, 1, 2, 3. the default is 255 if no value is present.
PdpAddress			V1/V2: IP Address length 16 ASN.1: IPV-6 Address	UE address on the TTG towards the GGSN. Same as “MT address” specified above in this table.

CDR Field Formats for Binary and ASN.1 CDR Files

Table 42 describes the field formats for binary Version 1 and Version 2 CDRs and Figure 25 on page 122 describes the field formats of the ASN.1 type formats.

Field Formats for Binary Version 1 and Binary Version 2 CDR Files

Table 42: Format Types for Binary Version 1 and Binary Version2 CDR Fields

Field Format	Description
Byte	Unsigned 8-bit integer in network byte order.
Integer	Unsigned 32-bit integer in network byte order.
Text	ASCII characters.
BCD	Binary-Coded-Decimal (BCD) with 0xF as padding. BCD format has the following characteristics: <ul style="list-style-type: none"> ■ packed ■ swapped nibbles ■ hexadecimal f padded
IP-Address	128-bit IPv6 Address network byte order, for Ipv4 addresses, the first 12 bytes are all 0x00.
Timestamp	Local time in binary Version 1 UTC time in binary Version 2

Field Formats for ASN.1 CDR Files

The contents of Figure 25 can also be found on the CD in the CallDetailRecordV2a.asn1 file in the Support_Files directory.

Figure 25: Format Types for ASN1 CDRs (extension cdr2a)

```

CallDetailsRecordV2a DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

CDRv2a ::= SEQUENCE {
    recordType                --[ 0]-- UInt8,
    servedIMSI                --[ 1]-- NumericString(SIZE(0..16)),
    chargingID                --[ 2]-- UInt32,
    gGSNIPAddress             --[ 3]-- IPv6-Address,
    nasAddress                 --[ 4]-- IPv6-Address,
    nASPortType               --[ 5]-- UInt32,
    nASTimezone               --[ 6]-- NAS-TimeZone,
    asAddress                  --[ 7]-- IPv6-Address,
    nodeID                    --[ 8]-- UTF8String(SIZE(0..20)),
    apName                     --[ 9]-- UTF8String(SIZE(0..63)),
    protocolType              --[10]-- Protocol-Type,
    mtAddress                  --[11]-- IPv6-Address,
    dataVolUplink              --[12]-- UInt32,
    dataVolDownlink           --[13]-- UInt32,
    recOpenTime                --[14]-- Timestamp,
    changeTime                 --[15]-- Timestamp,
    duration                   --[16]-- UInt32,
    causeRecClosing            --[17]-- UInt8,
    recSeqNum                  --[18]-- UInt32,
    chargingType               --[19]-- UInt8,
    chargingCharacteristics    --[20]-- UInt8,
    connectionType            --[21]-- UInt8,
    servedMSISDN              --[22]-- NumericString(SIZE(0..18)),
    cUIdType                   --[23]-- UInt8,
    cUIdLength                 --[24]-- UInt8,
    domainIndex                --[25]-- UInt8,
    cUId                       --[26]-- UTF8String(SIZE(0..253)),
    locationName               --[27]-- UTF8String(SIZE(0..128)),
    locationInfo               --[28]-- UTF8String(SIZE(0..253)),
    visitedOpID                --[29]-- UTF8String(SIZE(0..128)),
    operatorName               --[30]-- UTF8String(SIZE(0..128)),
    extChargingIdLen           --[31]-- UInt8,
    extChargingId              --[32]-- UTF8String(SIZE(0..253)),
    pDPChargingCharacteristics --[33]-- UInt8,
    qosTrafficClass            --[34]-- UInt8,
    pDPIPAAddress              --[35]-- IPv6-Address
}

UInt8 ::= INTEGER(0..255)          -- 0x00-0xff
UInt32 ::= INTEGER(0..4294967295) -- 0x00000000-0xffffffff
Timestamp ::= UInt32 -- seconds after 1970-01-01T00:00:00Z
-- (Unix epoch ~ constrained GeneralizedTime)
IPv6-Address ::= OCTET STRING(SIZE(16))
Protocol-Type ::= SEQUENCE { organization --[0]-- UInt8,
                             value       --[1]-- UInt8 }
NAS-TimeZone ::= SEQUENCE { tz15min --[0]-- UInt8,
                             dst     --[1]-- UInt8 }

END

```

Chapter 7

Configuring EAP-SIM/EAP-AKA Authentication

This chapter describes configuration tasks for using EAP/SIM or EAP/AKA credentials to authenticate mobile subscribers for wireless hotspot Internet access.

Configuring the `simauth.aut` File

The EAP-SIM/EAP-AKA module handles EAP-SIM authentication for clients using SIM cards and EAP-AKA authenticating for clients using USIM cards. The settings for the EAP-SIM/EAP-AKA module are stored in `simauth.aut` in the Steel-Belted Radius installation directory.



NOTE: Authenticating subscribers requires communication with the HLR or MSC. Follow the directions for “Configuring the `authGateway` Application for HLR Communication” on page 46 or “Configuring the `SMSGateway` Application for MSC Communication” on page 52 to establish connection with the HLR or MSC.

`simauth.aut` [Bootstrap] Section

The [Bootstrap] section of the `simauth.aut` file specifies information that Steel-Belted Radius uses to load and start the `simauth` module.

```
[Bootstrap]
LibraryName=simauth.so
Enable=1
InitializationString=EAP/SIM
```

Table 43: `simauth.aut` [Bootstrap] Fields

Field	Description
LibraryName	Specifies the name of the <code>simauth</code> module. Default value is <code>simauth.so</code> . Do not change this unless you are advised to do so by Juniper Technical Support.
Enable	If set to 1 (default), the <code>simauth</code> module is enabled. If set to 0, the <code>simauth</code> module is disabled. Default value is 1.

Table 43: simauth.aut [Bootstrap] Fields (continued)

Field	Description
InitializationString	This entry is used to specify the name of the simauth authentication method. Default value is EAP/SIM.

simauth.aut [Settings] Section

The [Settings] section of the simauth.aut file contains the following settings:

```
[Settings]
ConfigLog=ConsoleAndLog
EnableSIM=1
EnableAKA=1
NumberOfTriplets=3
ReauthenticationRealm=
ReauthenticationCountLimit=65535
ReauthenticationLifetimeSec=3600
PseudonymSecret=
PseudonymLifetimeDays=1
UseEAPResponseIdentity=0
EnableFailover=yes
FailoverTimeoutSec=60
ChargeableUserIdInResponse=IMSI
ChargeableUserIdAttribute=Chargeable-User-Identity
SendCUIDOnlyIfReceived=0
ProfileisUser=0
```



NOTE: For an overview of EAP-SIM/EAP-AKA pseudonyms and reauthentication identities, see “EAP-SIM/EAP-AKA Identities” on page 14.

Table 44: simauth.aut [Settings] Fields

Field	Description
ConfigLog	Specifies where simauth configuration information will be logged. Options are: <ul style="list-style-type: none"> ■ None – Do not log simauth configuration information. ■ Log – Record simauth configuration information in the Steel-Belted Radius log file. ■ ConsoleAndLog (default) – Record simauth configuration information in the Steel-Belted Radius log file and display configuration information on the Steel-Belted Radius console.
EnableEAPSIM	If set to 1, EAP-SIM authentication with GSM SIM cards is enabled. If set to 0, EAP-SIM authentication is disabled. Default value is 1 (enabled).
EnableEAPAKA	If set to 1, EAP-AKA authentication with 3G USIM cards is enabled. If set to 0, EAP-AKA authentication is disabled. Default value is 1 (enabled).

Table 44: simauth.aut [Settings] Fields (continued)

Field	Description
NumberOfTriplets (SIM only)	The number of triplets required for each authentication. The allowable values are 2 or 3. The higher the number of triplets, the more keying material is available and, consequently, the more the authentication is resistant to attacks. The default value is 3.
ReauthenticationRealm	The realm returned with a Reauthentication Identity that indicates where the return responses to reauthentication requests are directed. The default uses the realm from the Permanent Identity (if any exists) of the client.
ReauthenticationCount Limit	The number of allowed reauthentications before requesting fresh triplets and performing a complete authentication. The range is 0–65535. The default is 65535. If you enter 0, Reauthentication Identities are not generated.
ReauthenticationLifetime Sec	The duration (in seconds) of an identity that is generated by Steel-Belted Radius for the purpose of reauthentication. The default is 3600 (1 hour). The client must reauthenticate within this time or use a Pseudonym or Permanent Identity . Set to 0 to prevent reauthentication identities from being generated.
PseudonymSecret	A secret used to encrypt Pseudonyms . You can use any text string up to 32 characters. There is no default. If you do not specify a secret, pseudonyms are not generated. When you change this value, all pseudonyms assigned to currently authenticated clients are invalidated, requiring reauthentication.
PseudonymLifetimeDays	The lifetime of a Pseudonym in days. The default is 1 (day). The actual lifetime varies from the specified time, to twice the specified time.
UseEAPResponseIdentity	Set to 1 when the EAP-Response/Identity message is not altered by a proxy RADIUS server. If you set this to 1 and such changes do take place, then authentication fails. Set to 0 when the client EAP Identity is modified by a proxy RADIUS server. The default is 0.
EnableFailover	Set to yes to enable RADIUS failover by silent discard. Failover occurs when the HLR is inaccessible; further requests are silently discarded until the HLR becomes available.
FailoverTimeoutSec	If failover is enabled, the HLR will be presumed to be accessible after the specified number of seconds, and the next request will not be silently discarded. Default is 60 seconds. If you set this value to 0. HLR availability is not presumed and EAP-SIM/EAP-AKA authentication requests are discarded until the HLR responds.

Table 44: simauth.aut [Settings] Fields (continued)

Field	Description
ChargeableUserIdInResponse	<p>Specifies which identifier should be used as the CUID and returned in the Chargeable-User-Id (CUID) attribute in the RADIUS Access-Accept message.</p> <p>This field can be set to:</p> <ul style="list-style-type: none"> ■ None (default) ■ IMSI – The attribute format is 01:<i>imsi</i>, where <i>imsi</i> is the subscriber’s IMSI. ■ NAI – The attribute format is 02:<i>nai</i>, where <i>nai</i> is the subscriber’s NAI. ■ MSISDN –The attribute format is 03:<i>msisdn</i>, where <i>msisdn</i> is the subscriber’s MSISDN. <p>You can specify more than one CUID identifier. The identifiers must be comma separated. The identifier returned by the NAS will be used as the CUID. This identifier is usually the first identifier in the list.</p> <p>In the following example, the CUID in the CDR will be the IMSI.</p> <p>ChargeableUserIdInResponse=IMSI,NAI</p> <p>The IMSI and NAI values returned in the Access-Accept are derived from the User-Name attribute in the initial Access-Request. The MSISDN value returned in the Access-Accept is derived from the target module specified in the <code>gsmmap.gen</code> file. See “Configuring Each Realm Section” on page 61 and “Target Module Section” on page 64.</p>
ChargeableUserIdAttribute	<p>Specifies the attribute that will contain the CUID in the Access-Accept.</p> <p>This field can be set to either of these settings:</p> <ul style="list-style-type: none"> ■ Chargeable-User-Identity - The CUID will be returned in the Chargeable-User-Identity attribute. This setting complies with RFC 4372 available at http://www.ietf.org. ■ TeliaSonera-Chargeable-UserId - The CUID will be returned in the TeliaSonera-Chargeable-UserId attribute. This setting complies with GSM Association document IR6.1 available at http://www.gsmworld.com. <p>Note: The following call detail record (CDR) fields carry CUID information. See “CDR Fields” on page 113 for a list of all CDR types.</p> <p style="padding-left: 20px;">Charging ID</p> <p style="padding-left: 20px;">Domain Index</p> <p style="padding-left: 20px;">Charging Identifier</p>
SendCUIDOnlyIfReceived	<p>Used only if ChargeableUserIdAttribute is set to Chargeable-User-Identity.</p> <p>This field may be set to:</p> <ul style="list-style-type: none"> ■ 0 – The CUID attributes will be attached to the Access-Accept regardless of whether the CUID attribute was received in the Access-Request. ■ 1 – The CUID attribute will be attached to the Access-Accept only if the CUID was received in the Access-Accept.

***simauth.aut* [ProfileMap] Section**

Your HLR includes a database of subscribers. The database maps subscribers to the class(es) of service to which they are subscribed. You can configure the MAP Gateway to return strings that indicate a subscriber's service authorization. The service authorization strings returned by the MAP Gateway correspond to selected TS, BS, or ODB settings in the subscriber's profile on the HLR. For information about defining service authorization strings in the `authGateway.conf` file, see "Authorization Options" on page 48.

Alternatively, you can configure Steel-Belted Radius to request service authorization strings from an external SQL or LDAP database instead of from an HLR. For information about requesting authorization information from an SQL database, see "SQL Database Data Retrieval Methods" on page 72. For information about requesting authorization information from an LDAP directory, see "[Response] Section" on page 80.

You can create two types of Steel-Belted Radius entities that are used for specifying authorization policies:

- Profiles
- Native user

You can use Steel-Belted Radius profiles (or native users) to define classes of subscribers who are authorized for service. Refer to the *Steel-Belted Radius Administration Guide* for information about creating profiles (or native users).



NOTE: If you configure native users for use with Steel-Belted Radius/SIM Server, then you must modify `ProfileIsUser` in `simauth.aut`.

You can use the [ProfileMap] section of the `simauth.aut` file to assign one or more service authorization strings to these Steel-Belted Radius profiles (or native users). By doing so, each time that a subscriber requests an account, the service that is specified by the HLR is checked against the strings that you assign to a profile or native user in the [ProfileMap] section of the `simauth.aut`:

- If the set of service authorization strings do not match those of any profile or native user who you list, the provisioning request is denied.
- The profile map entries are checked in order. When the set of service strings match those of a profile or native user who you list, the subscriber is assigned that Steel-Belted Radius profile or native user.
- Although the authorization string values must match those in `authGateway.conf`, you can include authorization strings that are valid for multiple HLRs in a given line corresponding to a profile or native user in the [ProfileMap] section of `simauth.aut`.

You can use each line in the [ProfileMap] section of `simauth.aut` to provide a set of HLR authorization strings (from `authGateway.conf`) that specify an authorization policy defined by a Steel-Belted Radius profile or native user.

The format for each line is as follows:

```
ProfileName1=auth1:auth2:auth3
```

where `ProfileName1` is the name of a Steel-Belted Radius profile or native user, and `auth1`, `auth2`, and `auth3` are the HLR authorization strings configured in the MAP Gateway `authGateway.conf` file. The sequence and capitalization of the authorization strings are ignored. You can specify up to 128 authorization strings.

To use a particular profile or native user to define an authorization policy, make sure all the listed authorization strings exactly match the authorization strings returned from the MAP Gateway. You can also use a wildcard `*` to match any otherwise unmatched strings. For example, you can have the following:

```
ProfileName1=auth1:auth2:*
```

This matches any set of authorization strings as long as they include `auth1` and `auth2`.

You can specify combinations of authorization strings for which authorization is denied by entering one or more DENY lines. A DENY line is of the form:

```
<DENY>=auth1:auth2:auth3
```

where the profile name is not specified and where `auth1`, `auth2`, and `auth3` are the HLR authorization strings configured in the MAP Gateway `authGateway.conf` file. The same matching and wildcard rules apply as with PROFILE lines. For example, the following statement denies authorization if the authorization strings returned are `auth1`, `auth2`, and `auth3`.

```
<DENY>=auth1:auth2:auth3
```

The following statement denies authorization if the authorization strings returned include both `auth1` and `auth2` and any other strings.

```
<DENY>=auth1:auth2:*
```

Similarly, the following statements deny authorization if the authorization strings returned include `auth1` or `auth2`.

```
<DENY>=auth1:*
```

```
<DENY>=auth2:*
```

Finally, the following statement denies authorization if no authorization strings are returned:

It is often useful to place DENY statements at the top of the list because a subscriber is assigned the policy associated with the first match in the list of profiles (or native users) that you include in the profile map. Place PROFILE statements with specific criteria in the middle of the list and PROFILE statements with wildcards at the bottom of the list.



NOTE: If the set of service authorization strings do not match those of any profile or native user who you list, the provisioning request is denied.

Chapter 8

Configuring Web-Based Authentication with SMS

SMS (Short Message Service) provides a secure method for provisioning passwords to public WLAN access through a RADIUS/IP-based public network. The subscriber's temporary password is transmitted securely as a text message to the subscriber's device using SMS.



NOTE: Authenticating subscribers requires communication with the HLR or MSC. Follow the directions for “Configuring the authGateway Application for HLR Communication” on page 46 or “Configuring the SMSGateway Application for MSC Communication” on page 52 to establish connection with the HLR or MSC.

Configuring SMS Options

You can configure SMS options in Steel-Belted Radius options by modifying the files listed in Table 45.

Table 45: SMS Options in Steel-Belted Radius/SIM Server

File	Options
radius.ini	Steel-Belted Radius options
smsGateway.conf	SMS Gateway routing configuration
smsprov.aut	Account provisioning options
smsauth.aut	SMS authentication options
cdراعct.acc	Accounting options
ss7ldapdb.gen	LDAP database settings
gsmmap.gen	Ulticom MAP Gateway settings
smsmsg.gen	SMS messages

Configuring Account Provisioning Options

The settings for SMS provisioning are stored in `smsprov.aut` in the Steel-Belted Radius installation directory.

`smsprov.aut` [Bootstrap] Section

The [Bootstrap] section of the `smsprov.aut` file contains the following settings:

```
[Bootstrap]
LibraryName=smsprov
Enable=1
InitializationString=smsprov
```

Table 46: `smsprov.aut` [Bootstrap] Fields

Field	Description
LibraryName	The name of the library called when <code>smsprov</code> runs. The default value is <code>smsprov</code> .
Enable	Set to 1 to enable the features described in this file. Set to 0 to disable the features described in this file. The default is 1.
InitializationString	The <code>smsprov</code> initialization string. The default is <code>smsprov</code> .

`smsprov.aut` [Settings]

The [Settings] section of the `smsprov.aut` file contains the following:

```
[Settings]
ConfigLog=ConsoleAndLog
ProvisionedUnusedLifetimeSecs=3600
MaxPasswordResends=1
MinAcctProvisionRequestSecs=3600
MaxAcctProvisionRequestSecs=28800
DefaultAcctProvisionRequestSecs=14400
PasswordLength=10
```

Table 47: `smsprov.aut` [Settings] Fields

Field	Description
ConfigLog	Specifies where configuration information will be logged. Options are: <ul style="list-style-type: none"> ■ None – Do not log configuration information. ■ Log – Record configuration information in the Steel-Belted Radius log file. ■ ConsoleAndLog (default) – Record configuration information in the Steel-Belted Radius log file and display configuration information on the Steel-Belted Radius console.
ProvisionedUnusedLifetimeSecs	The number of seconds after which an unused temporarily provisioned account is deleted. The default value is 3600 (one hour).

Table 47: smsprov.aut [Settings] Fields (continued)

Field	Description
MaxPasswordResends	The maximum number of times that a temporary password is retransmitted to the subscriber's cell phone for a given provisioned account. The default value is 1.
MinAcctProvisionRequestSecs	The minimum duration of a provisioned account. You might need to configure this when you offer different classes of service. Any service requests for accounts that last less than this value are rejected. The default is 3600 (1 hour).
MaxAcctProvisionRequestSecs	The maximum duration of a provisioned account. You might need to configure this when you offer different classes of service. The default is 28800 (8 hours). Any service requests for accounts for an amount of time that exceeds this value are assigned this maximum duration.
DefaultAcctProvisionRequestSecs	The duration of any fixed-fee provisioned account when no specific duration is present in the provisioning request. The default value is 14400 (4 hours).
PasswordLength	The number of characters required for the provisioned password. The default value is 10.
AuthGatewayBarAccessIndicator	Text value assigned to subscribers who are designated as barred (ODB) and who are not permitted service by the HLR. The value entered here must match the value entered in the <code>authGateway.conf</code> file. Default value is bar.

smsprov.aut [ProfileMap] Section

Your HLR includes a database of subscribers. The database maps subscribers to the class(es) of service to which they are subscribed. You can configure the MAP Gateway to return strings that indicate a subscriber's service authorization. The service authorization strings returned by the MAP Gateway correspond to selected TS, BS, or ODB settings in the subscriber's profile on the HLR. For information about defining service authorization strings in the `authGateway.conf` file, see "Authorization Options" on page 48.

Alternatively, you can configure Steel-Belted Radius to request service authorization strings from an external SQL or LDAP database instead of from an HLR. For information about requesting authorization information from an SQL database, see "SQL Database Data Retrieval Methods" on page 72. For information about requesting authorization information from an LDAP directory, see "[Response] Section" on page 80.

You can create two types of Steel-Belted Radius entities that are used for specifying authorization policies:

- Profiles
- Native user

You can use Steel-Belted Radius profiles (or native users) to define classes of subscribers who are authorized for service. Refer to the *Steel-Belted Radius Administration Guide* for information about creating profiles (or native users).



NOTE: If you configure native users for use with Steel-Belted Radius/SIM Server, you must modify `ProfileIsUser` in `smsauth.aut`. See “`smsauth.aut` [Settings] Section” on page 134.

You can use the `[ProfileMap]` section of the `smsprov.aut` file to assign one or more service authorization strings to these Steel-Belted Radius profiles (or native users). By doing so, each time that a subscriber requests an account, the service that is specified by the HLR is checked against the strings that you assign to a profile or native user in the `[ProfileMap]` section of the `smsprov.aut`:

- If the set of service authorization strings do not match those of any profile or native user that you list, the provisioning request is denied.
- The profile map entries are checked in order. When the set of service strings match those of a profile or native user that you list, the subscriber is assigned that Steel-Belted Radius profile or native user.
- Although the authorization string values must match those in `authGateway.conf`, you can include authorization strings that are valid for multiple HLRs in a given line corresponding to a profile or native user in the `[ProfileMap]` section of `smsprov.aut`.

You can use each line in the `[ProfileMap]` section of `smsprov.aut` to provide a set of HLR authorization strings (from `authGateway.conf`) that specify an authorization policy defined by a Steel-Belted Radius profile or native user.

The format for each line is as follows:

```
ProfileName1=auth1:auth2:auth3
```

where `ProfileName1` is the name of a Steel-Belted Radius profile or native user, and `auth1`, `auth2`, and `auth3` are the HLR authorization strings configured in the MAP Gateway `authGateway.conf` file. The sequence and capitalization of the authorization strings are ignored. You can specify up to 128 authorization strings.

To use a particular profile or native user to define an authorization policy, make sure that all the listed authorization strings exactly match the authorization strings returned from the MAP Gateway. You can also use a wildcard `*` to match any otherwise unmatched strings. For example, the following setting matches any set of authorization strings as long as they include `auth1` and `auth2`:

```
ProfileName1=auth1:auth2:*
```

You can specify combinations of authorization strings for which authorization is denied by entering one or more DENY lines. A DENY line is of the form:

```
<DENY>=auth1:auth2:auth3
```

where the profile name is not specified and where `auth1`, `auth2`, and `auth3` are the HLR authorization strings configured in the MAP Gateway `authGateway.conf` file. The same matching and wildcard rules apply as with PROFILE lines. For example, the following statement denies authorization if the authorization strings returned are `auth1`, `auth2`, and `auth3`.

```
<DENY>=auth1:auth2:auth3
```

The following statement denies authorization if the authorization strings returned include both `auth1` and `auth2` and any other strings.

```
<DENY>=auth1:auth2:*
```

Similarly, the following statements deny authorization if the authorization strings returned include `auth1` or `auth2`.

```
<DENY>=auth1:*
```

```
<DENY>=auth2:*
```

Finally, the following statement denies authorization if no authorization strings are returned:

```
<DENY>=
```

Because a subscriber is assigned the policy associated with the first match in the list of profiles (or native users) that you include in the profile map, it is often useful to place DENY statements at the top of the list, PROFILE statements with specific criteria in the middle of the list, and PROFILE statements with wildcards at the bottom of the list.



NOTE: If the set of service authorization strings do not match those of any profile or native user that you list, the provisioning request is denied.

Configuring SMS Authentication Options

The settings for SMS authentication are stored in the `smsauth.aut` file in the Steel-Belted Radius installation directory.

smsauth.aut [Bootstrap] Section

The [Bootstrap] section of the `smsauth.aut` file contains the following settings.

```
[Bootstrap]
LibraryName=smsauth.so
Enable=1
InitializationString=SMSAUTH
```

Table 48: smsauth.aut [Bootstrap] Fields

Field	Description
LibraryName	The name of the library called when smsprov runs. The default value is <code>smsauth.so</code> .
Enable	Set to 1 to enable the features described in this file. Set to 0 to disable the features described in this file. The default is 1.
InitializationString	The <code>smsauth</code> initialization string. The default is <code>SMSAUTH</code> .

smsauth.aut [Settings] Section

The [Settings] section of the `smsauth.aut` file contains the following settings.

```
[Settings]
ConfigLog=ConsoleAndLog
ProfileIsUser=0
ValidateStationID=0
SendMsgBeforeExpiresSec=300
NumExpireSoonMessageThreads=100
```

Table 49: smsauth.aut [Settings] Fields

Field	Description
ConfigLog	Specifies where configuration information will be logged. Options are: <ul style="list-style-type: none"> ■ None – Do not log configuration information. ■ Log – Record configuration information in the Steel-Belted Radius log file. ■ ConsoleAndLog (default) – Record configuration information in the Steel-Belted Radius log file and display configuration information on the Steel-Belted Radius console.
ProfileIsUser	Specifies whether the authorization policy specifiers listed in the [ProfileMap] section of <code>smsprov.aut</code> represent Steel-Belted Radius native users or profiles. <ul style="list-style-type: none"> ■ 0 (default) – Steel-Belted Radius profiles. ■ 1 – Steel-Belted Radius native users.
ValidateStationID	Specifies whether Steel-Belted Radius/SIM Server stores the MAC address of the subscriber's computer (forwarded by the remote AS) and matches that MAC address to subsequent authentication requests during the subscriber's session. <ul style="list-style-type: none"> ■ If set to 0, do not perform MAC address comparison. ■ If set to 1, perform MAC address comparison during re-authentication and reject authentication requests when the submitted MAC address does not match the stored MAC address. <p>Note that the remote AS must support forwarding of the subscriber's MAC address in the Calling-Station-Id attribute for MAC address comparison to work.</p>

Table 49: smsauth.aut [Settings] Fields (continued)

Field	Description
SendMsgBeforeExpiresSec	Specifies the threshold (in seconds) at which an “about to expire” SMS message is sent to a subscriber using a temporary account. Default is 300 seconds. Note: The <code>smsmsg.gen</code> file controls whether “about to expire” messages are sent and specifies the format of the “about to expire” message. For more information, see “Configuring SMS Message Options.”
NumberExpireSoonMessageThreads	Specifies the number threads that can be used to send “about to expire” SMS messages to users. Default is 100. A warning message is written to the Steel-Belted Radius log file if a message is scheduled to be sent but a thread is not available. Increase the value of this parameter if warning messages appear regularly in your log file.

Configuring SMS Message Options

The `smsmsg.gen` file contains settings to indicate options about the text message that will be sent to the subscriber.

[Bootstrap] Section

The [Bootstrap] section of the `smsmsg.gen` file contains the following settings:

```
[Bootstrap]
LibraryName=smsmsg
Enable=1
DependsOn=gsmmap.gen
```

Table 50: smsmsg.gen [Bootstrap] Fields

Field	Description
LibraryName	Specifies the name of the SMS message library. Default value is <code>smsmsg</code> .
Enable	Set to 1 to enable SMS messages. Set to 0 to disable SMS messages. Default value is 1.
DependsOn	Specifies the module or modules that must be running for SMS messaging to work. Default value is <code>gsmmap.gen</code> .

[Settings] Section

You can configure the [Settings] section of `smsmsg.gen` to specify the language and text for SMS messages. You can create messages in multiple languages. In addition, you can specify the text for password response messages, password reminder messages, and elapsed time messages.

```
[Settings]
ConfigLog=ConsoleAndLog
```

```

SendPasswordReminder=1
SendWelcome=1
SendEndTimeRemaining=1
SendEndTimeExpired=1
SendExpiringSoon=1
DefaultLanguageCode=en
LanguageCodeSectionsList=en,fi,sv

```

Table 51: smsmsg.gen [Settings] Fields

Field	Description
ConfigLog	<p>Specifies where SMS message configuration information will be logged. Options are:</p> <ul style="list-style-type: none"> ■ None – Do not log SMS message configuration information. ■ Log – Record SMS message configuration information in the Steel-Belted Radius log file. ■ ConsoleAndLog (default) – Record SMS message configuration information in the Steel-Belted Radius log file and display configuration information on the Steel-Belted Radius console.
DefaultLanguageCode	<p>Specifies the ISO-639 code identifying the default language to use for SMS text messages when a language is not specified in a provisioning request.</p> <p>The default value is en (English).</p>
LanguageCodeSectionsList	<p>A comma-separated list of ISO-639 codes for the languages that you intend to use for SMS messages. For example, you can use en,fr for English and French.</p> <p>The default value is en,fi,sv (English, Finnish, Swedish).</p>
SendExpiringSoon	<p>Enter 1 (default) to enable “session ending soon” SMS text to be sent when the time remaining on a user’s session falls below a configurable time threshold.</p> <p>Enter 0 to stop password reminders from being sent.</p> <p>If value is 1, you should specify a message string for ExpiringSoonTemplate in each language section in this file.</p> <p>Note: The smsauth.aut file specifies the time threshold used to trigger “about to expire” messages. For more information, see page 135.</p>
SendPasswordReminder	<p>Enter 1 (default) to enable a password reminder text to be sent.</p> <p>Enter 0 to stop password reminders from being sent.</p> <p>If value is 1, you should specify a message string for PasswordReminderTemplate in each language section in this file.</p>
SendWelcome	<p>Enter 1 (default) to enable a welcome message to be sent.</p> <p>Enter 0 to indicate a welcome message should not be sent.</p> <p>If value is 1, you should specify a message string for WelcomeTemplate in each language section in this file.</p>
SendPasswordReminder	<p>Enter 1 (default) to allow password reminder text to be sent.</p> <p>Enter 0 to stop password reminders from being sent.</p>
SendEndTimeRemaining	<p>Enter 1 (default) to enable a message to be sent identifying how much time remains in the user’s session at termination.</p> <p>Enter 0 to stop time reminders from being sent.</p> <p>If value is 1, you should specify a message string for EndTimeRemainingTemplate in each language section in this file.</p>

Table 51: smsmsg.gen [Settings] Fields (continued)

Field	Description
SendEndTimeExpired	Enter 1 (default) to allow a message informing a user that a network session has expired. Enter 0 to stop time reminders from being sent. If value is 1, you should specify a message string for EndTimeExpiredTemplate in each language section in this file.

Setting Up Message Templates

You must include separate sections for message templates in each language that you list for LanguageCodeSectionsList. The title for the section must be the code identifying that language (for example, en for English or fi for Finnish) in brackets.

[en]

PasswordMessageTemplate = Your authentication password is <P>. Enter it on the login web page.

PasswordReminderTemplate = Password has already been sent to you. Your authentication password is <P>. Enter it on the login web page.

WelcomeTemplate = You have logged in to the network. You have <T> minutes connection time.

EndTimeRemainingTemplate = You have been logged out of the WLAN network. You have <T> minutes connection time left.

EndTimeExpiredTemplate = Your WLAN subscription has no more connection time left. The system has logged you out of the WLAN network.

ExpiringSoonTemplate = You have <T> minutes WLAN connection time left. Please, prepare for the connection closure! You can order more time via Internet.

Table 52: smsmsg.gen [Language] Fields

Field	Description
PasswordMessageTemplate	A string consisting of text and variables for the message sent with a user's temporary password when it is first issued. You can assign different message values to this variable under different language sections of this file. You can use the following coded strings (in brackets) to represent subscriber-dependent content: <P>: Password <R>: Realm <M>: MSISDN <T>: Time remaining You cannot include hard returns in messages. Default value for English is Your authentication password is <P>. Enter it on the login web page.

Table 52: smsmsg.gen [Language] Fields (continued)

Field	Description
PasswordReminderTemplate	<p>A string consisting of text and variables for the message sent when someone is logging back in to an active account. You can assign different message values to this variable under different language sections of this file.</p> <p>You can use the following coded strings (in brackets) to represent subscriber-dependent content:</p> <p><P>: Password <R>: Realm <M>: MSISDN <T>: Time remaining</p> <p>You cannot include hard returns in messages.</p> <p>Default value for English is Password has already been sent to you. Your authentication password is <P>. Enter it on the login web page.</p>
WelcomeTemplate	<p>A string consisting of text and variables for the message sent when someone is logging into the network. You can assign different message values to this variable under different language sections of this file.</p> <p>You can use the following coded strings (in brackets) to represent subscriber-dependent content:</p> <p><P>: Password <R>: Realm <M>: MSISDN <T>: Time remaining</p> <p>You cannot include hard returns in messages.</p> <p>Default value for English is You have logged in to the network. You have <T> minutes connection time.</p>
EndTimeRemainingTemplate	<p>A string consisting of text and variables for the message sent to signal how much time is left in a fixed-time session that a subscriber is terminating before it expires. You can assign different message values to this variable under different language sections of this file.</p> <p>You can use the following coded strings (in brackets) to represent subscriber-dependent content:</p> <p><P>: Password <R>: Realm <M>: MSISDN <T>: Time remaining</p> <p>You cannot include hard returns in messages.</p> <p>Default value for English is You have been logged out of the WLAN network. You have <T> minutes connection time left.</p>

Table 52: smsmsg.gen [Language] Fields (continued)

Field	Description
EndTimeExpiredTemplate	<p>A string consisting of text and variables for the message sent to signal that a user session has expired. You can assign different message values to this variable under different language sections of this file.</p> <p>You can use the following coded strings (in brackets) to represent subscriber-dependent content:</p> <p><P>: Password <R>: Realm <M>: MSISDN <T>: Time remaining</p> <p>You cannot include hard returns in messages.</p> <p>Default value for English is Your WLAN subscription has no more connection time left. The system has logged you out of the WLAN network.</p>
ExpiringSoonTemplate	<p>A string consisting of text and variables for the message sent to signal that a user's session will expire in the immediate future. You can assign different message values to this variable under different language sections of this file.</p> <p>You can use the following coded strings (in brackets) to represent subscriber-dependent content:</p> <p><P>: Password <R>: Realm <M>: MSISDN <T>: Time remaining</p> <p>You cannot include hard returns in messages.</p> <p>Default value for English is You have <T> minutes WLAN connection time left. Please, prepare for the connection closure! You can order more time using the Internet.</p>

SMS Interface Requirements

This section describes the interface requirements for the interaction between the Access Controller and Steel-Belted Radius/SIM Server.

Overview

The Access Controller uses the RADIUS interface requirements described in this section to provision the temporary subscriber account and one-time password. The MSISDN of the subscriber's mobile device is the username for the temporary account. The one-time password is generated by SIM Server and is communicated to the user with an SMS message sent to the subscriber's mobile device.

Once the temporary account and password have been provisioned then the Access Controller uses conventional RADIUS authentication and accounting procedures, as with any subscriber account, to manage the subscriber's connection to the Internet. For the typical Access-Request, the username is the MSISDN and the password is the one-time password assigned to the temporary account.

Access-Request

The provisioning Access-Request must have a username and a password in a specific format. The provisioning reply is always an Access-Reject. The Reply-Message attribute attached to the Access-Reject contains the actual provisioning status.

Username Format

The username must be in the form: MSISDN@realm. That same format can be shown as:

MSISDN@realm

The MSISDN is composed of an optional + character followed by 8 to 18 digits in ASCII format.

For example, the following lines are both valid usernames:

+8567872785@provider.com

8567872785@provider.com

Typically, the subscriber enters only the MSISDN into the Web page and the Access Controller itself adds the @*realm* section of the username.

Password Format

Only Password Authentication Protocol (PAP) passwords are accepted.

The password must be in the form:

SmsProvision_T:duration_L:language.

where:

SmsProvision is case-sensitive

duration specifies the maximum duration of the temporary account and its associated one-time password. The *duration* field must consist of ASCII digits and is in units of seconds. (Optional field.)

If *duration* is not specified, the default value is used. The default value is the value of `DefaultAcctProvisionRequestSecs` in the `smsprov.aut` field.

language is one of the two-character lower-case ASCII alphabetic language codes specified in ISO-639. (Optional field.)

If the *language* is not specified, the default value is used. The default value is the value of `DefaultAccountProvisionRequestSeconds` in the `smsprov.aut` field.

The following are examples of valid passwords:

```
SmsProvision_T:3600_L:en
SmsProvision_L:en_T:3600
SmsProvision
SmsProvision_L:mn
SmsProvision_T:7200
```

Access-Request

The Access-Reject is used for both successful and failed provisioning. (The Access-Accept message is not used during provisioning.)

The provisioning status is carried in the Reply-Message attribute of Access-Reject message. The Reply-Message carries an ASCII string as shown in the following example from the C code.

Figure 26: Example C Code Showing ASCII String in Reply-Message

```
//
// The SMS error codes as strings.
//
#define SMS_SUCCESSFUL_STRING "0 - Successful"
#define SMS_NO_ROUTE_TO_HOME_NETWORK_STRING "1024 - No route to home network"
#define SMS_NON_EXISTENT_SUBSCRIBER_STRING "1025 - Non-existent subscriber"
#define SMS_CALL_BARRING_ON_STRING "1026 - Call barring on"
#define SMS_AUTHENTICATION_FAILED_STRING "1030 - Authentication failed"
#define SMS_NO_SERVICE_SUBSCRIBED_STRING "1031 - No service subscribed"
#define SMS_AUTHENTICATION_NETWORK_ERROR_STRING "1050 - Authentication network error"
#define SMS_SMS_MEMORY_FULL_STRING "2001 - SMS memory full"
#define SMS_MOBILE_TERMINAL_UNREACHABLE_STRING "2002 - Mobile terminal unreachable"
#define SMS_SUBSCRIBER_ALREADY_PROVISIONED_STRING "2003 - Subscriber already provisioned"
#define SMS_SESSION_TIMED_OUT_STRING "2004 - Session timed out"
#define SMS_PASSWORD_SHORT_MESSAGE_ALREADY_SENT_STRING "2005 - Password short message already sent"
```

Success String

Table 53: smsmsg.gen [Language] Fields

Text in Reply-Message	Description
0 - Successful	Indicates a successful provisioning (although sent in Access-Reject).

Failure Strings

Table 54: smsmsg.gen [Language] Fields

Text in Reply-Message	Description
1024 - No route to home network	User's home operator could not be reached due to missing IMSI analysis or SS7 network configuration.
1025 - Non-existent Subscriber	MSISDN could not be solved to an existing subscriber.
1026 - Call barring on	User's outgoing calls are barred in HLR profile.
1031 - No service subscribed	The user does not subscribe WLAN service.
1050 - Authentication network error	Internal error in SIM Server or MAP gateway.
2001 - SMS memory full	There's no free memory in user's mobile phone for short message.
2002 - Mobile terminal unreachable	User's mobile phone could not be reached for delivering a short message. Phone is out of coverage area or switched off.
2003 - Subscriber already provisioned	Maximum provisioning attempt limit reached for the subscriber; no short message is sent.

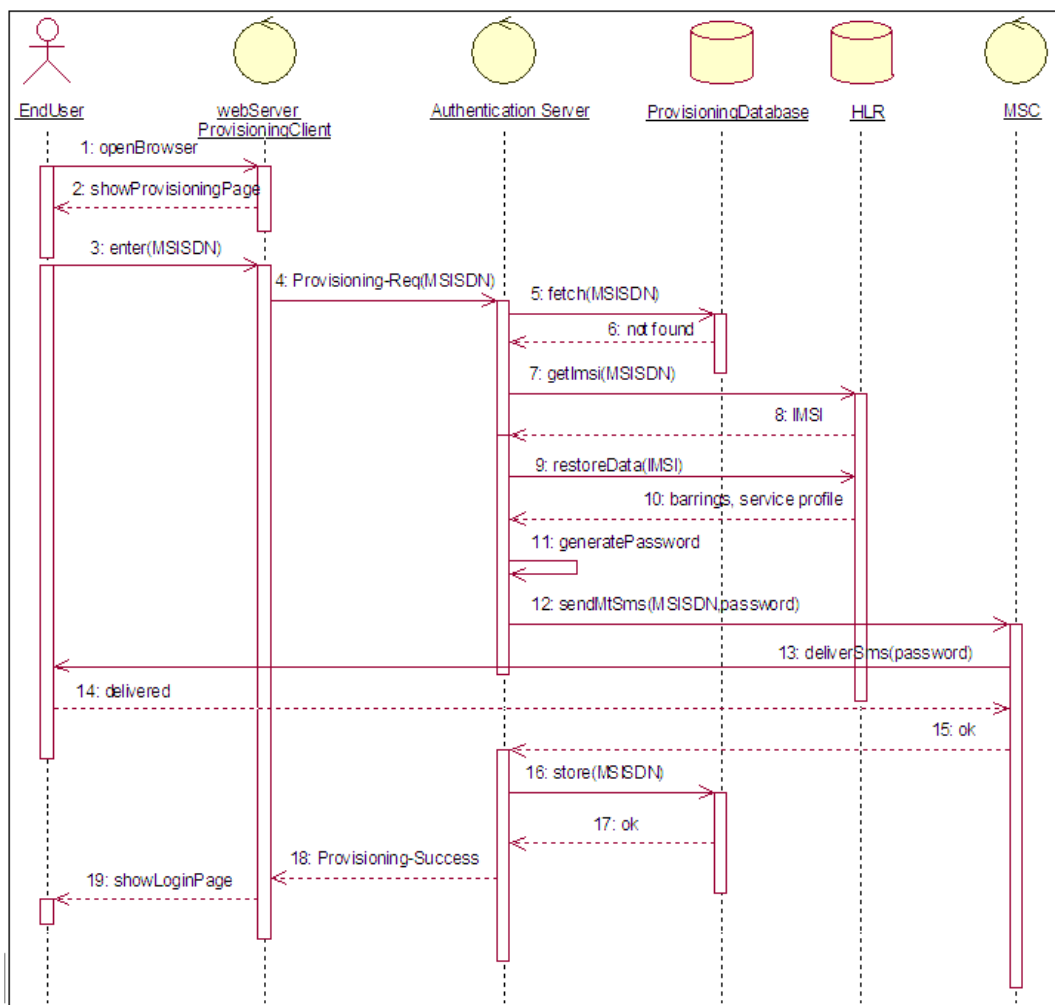
Interaction Examples

The following examples show the interaction between devices involved in the provisioning.

Example 1: Successful Provisioning When No Temporary Account Exists

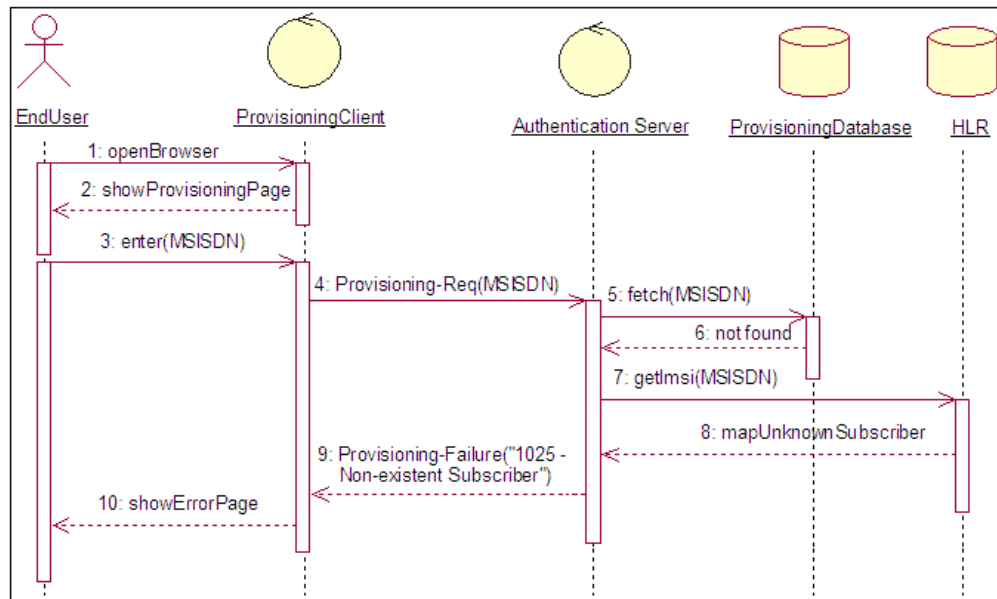
In Example 1, a user, who is not previously known in the system, is provisioned. The user has a postpaid subscription and has the SLAN service activated. A password is sent to the mobile device in an SMS message. The Web server displays the password dialog page.

Figure 27: Successful Provisioning When No Temporary Account Exists



Example 2: Failed Provisioning Due to Unknown MSISDN

In Example 2, the user is trying to provision with a random phone number. The MSISDN entered by the user does not belong to a known subscriber. The customer is informed with the Internet that the phone number is not valid.

Figure 28: Failed Provisioning Due to Unknown MSISDN

Appendix A

SS7 Sample Configuration Files

This appendix shows the sample configuration files for creating basic SS7 connection to an HLR. These sample files show how to set up one board with two SS7 links.

Basic Provisioning MML File with One Point Code/ Two SS7 Links

```
CRTE-OSPC:PC=9619,NI=NATO;  
CRTE-LSET:LSET=LSET1,PC=6400;
```

Local Signaling Point Code (in decimal format) using national rules.

SPC of SS7 GW where the link is connected.

```
CRTE-SLK:SLK=LNK1,LSET=LSET1,SLC=0,SPEED=64K,PORT=0,CHANNEL=2;  
CRTE-SLK:SLK=LNK2,LSET=LSET1,SLC=1,SPEED=64K,PORT=1,CHANNEL=3;
```

Define two SS7 links (channel=timeslot+1).

```
CRTE-RSET:RSET=RSET1,PC=6400,RTES=LSET1,LOADSHR=YES;
```

Define route to SS7 GW.

```
ALW-RSET:RSET=RSET1;  
ACTV-SLK:SLK=LNK1;  
ACTV-SLK:SLK=LNK2;
```

Activate route and links.

```
CREATE-GT:TT=0,NP=ISDN-TEL,NA=INT,DIG="35840211430",PC=9619,SSN=0,RI=GT  
;
```

```
CREATE-GT:TT=0,NP=ISDN-TEL,NA=INT,DIG="358",PC=6400,SSN=0,RI=DEF;  
CREATE-GT:TT=0,NP=ISDN-MOB,NA=INT,DIG="358",PC=6400,SSN=0,RI=DEF;
```

Define own GT number and point code for routing. (Routing based on GT.)

Notes:

TT — usually 0.

NP — 1=E.164=ISDN-TEL; 7=E.214=ISDN-MOB
 NA — INT
 DIG — Enter the first digits of MSISDN/IMSI. The routing decision is made with “best match” method. The PC message is sent towards this Point Code if “digit” matches the item specified as NP (ISDN-Tel).
 Point codes — in decimal format.

Routing Based on PC/SSN

CRTE-OSPC:PC=9619,NI=NAT1;

CRTE-LSET:LSET=LSET1,PC=11;

CRTE-SLK:SLK=LNK1,LSET=LSET1,SLC=0,SPEED=64K,PORT=0,CHANNEL=2;
 CRTE-SLK:SLK=LNK2,LSET=LSET1,SLC=1,SPEED=64K,PORT=1,CHANNEL=3;

CRTE-RSET:RSET=RSET1,PC=11,RTES=LSET1,LOADSHR=YES;
 CRTE-RSET:RSET=RSET2,PC=17,RTES=LSET1,LOADSHR=YES;
 CRTE-RSET:RSET=RSET3,PC=18,RTES=LSET1,LOADSHR=YES;

Define route for gateway and for each HLR.

CRTE-REMSSN:PC=17,SSN=6;
CRTE-REMSSN:PC=18,SSN=6;

Attach SSN and PC numbers for PCSSN routing.

ALW-RSET:RSET=RSET1;
 ALW-RSET:RSET=RSET2;
 ALW-RSET:RSET=RSET3;

ACTV-SLK:SLK=LNK1;
 ACTV-SLK:SLK=LNK2;

Inbound GT analysis. One for SIM Server and one for each HLR.

CREATE-GT:TT=0,NP=ISDN-TEL,NA=INT,DIG="35840299",**PC=961,SSN=7,RI=PCSSN**;
 CREATE-GT:TT=0,NP=ISDN-TEL,NA=INT,DIG="358402111",**PC=17,SSN=6,RI=PCSSN**;
 CREATE-GT:TT=0,NP=ISDN-TEL,NA=INT,DIG="358402112",**PC=18,SSN=6,RI=PCSSN**;

Outbound GT analysis. IMSI ranges go to a specific HLR.

CREATE-GT:TT=0,NP=ISDN-TEL,NA=INT,DIG="244911",**PC=17,SSN=6,RI=PCSSN**;
 CREATE-GT:TT=0,NP=ISDN-TEL,NA=INT,DIG="244912",**PC=18,SSN=6,RI=PCSSN**;
 CREATE-GT:TT=0,NP=ISDN-TEL,NA=INT,DIG="244913",**PC=18,SSN=6,RI=PCSSN**;
 CREATE-GT:TT=0,NP=ISDN-TEL,NA=INT,DIG="244914",**PC=17,SSN=6,RI=PCSSN**;

Redundant SS7 Links Backing Up Each Other (Two Point Codes)

CRTE-OSPC:PC=9619,NI=NATO;
 CRTE-LSET:LSET=**LSET1,PC=6400**;
 CRTE-LSET:LSET=**LSET2,PC=6401**;

Define two link sets using different point codes.

CRTE-SLK:SLK=LNK1,LSET=**LSET1**,SLC=0,SPEED=64K,PORT=0,CHANNEL=2;
 CRTE-SLK:SLK=LNK2,LSET=**LSET2**,SLC=1,SPEED=64K,PORT=1,CHANNEL=3;
 CRTE-SLK:SLK=LNK3,LSET=**LSET1**,SLC=2,SPEED=64K,PORT=**16**,CHANNEL=2;
 CRTE-SLK:SLK=LNK4,LSET=**LSET2**,SLC=3,SPEED=64K,PORT=**17**,CHANNEL=3;

```

CRTE-RSET:RSET=RSET1,PC=6400,RTES=LSET1&LSET2;
CRTE-RSET:RSET=RSET2,PC=6401,RTES=LSET2&LSET1;

ALW-RSET:RSET=RSET1;
ALW-RSET:RSET=RSET2;

ACTV-SLK:SLK=LNK1;
ACTV-SLK:SLK=LNK2;
ACTV-SLK:SLK=LNK3;
ACTV-SLK:SLK=LNK4;

CRTE-GT:TT=0,NP=ISDN-TEL,NA=INT,DIG="358402114300",PC=9619,SSN=0,RI=GT;
CRTE-GT:TT=0,NP=ISDN-TEL,NA=INT,DIG="358",PC=6400,SSN=0,RI=GT,
BKUPPC=6401,BKUPSSN=0,BKUPRI=GT;

```

Primary route for PC6400 is LSET1.
Backup route is LSET2.

Attach links from one point code to different links to maximize the redundancy.

authGateway Commands and Files

AS4StartMapGw.mml — One Point Code/ Two SS7 Links

```

CRTE-PROCESS:NAME="GMT",CE="as1",EXEC="authGateway -name GMT -port 2001
-host as1 -node MGW -prot C7 -conf conf/authGateway.conf.100 -lri 0 -lgti 4 -lssn 7 -ltp
0 -lnp 1 -lnai 4 -appctx 2 -ldigits 358402114300";

```

```

START-PROCESS:NAME="GMT", CE="as1";

```

AS4StartMapGw.mml — Routing Based on PC/SSN

```

CRTE-PROCESS:NAME="GMT",CE="as1",EXEC="authGateway -name GMT -port 2001
-host as1 -node MGW -prot C7 -conf conf/authGateway.conf.100 -lri 1 -lssn 7 -appctx 2
-ldigits 358402114300"; START-PROCESS:NAME="GMT", CE="as1";

```

Table 55: Parameters Used in Create and Start Commands

Parameter	Description
name	Process name
port	Remote port specified in ulcmmg.conf
host	Hostname
node	Node name (used in swmml command)
prot	Variant used (C7, A7 or CH7). C7 in Europe
conf	Location of authGateway.conf file
lri	Routing indicator. 0 = GT, 1 = PC/SSN
lgti	(GT only) Local GTI value, 4 for C7 network; 2 for A7
lssn	Local Subsystem Number (SSN)
lnp	1 = ISDN/Telephony

Table 55: Parameters Used in Create and Start Commands (continued)

Parameter	Description
lnai	Nature of Address Indicator. (4 = INT)
appctx	MAP version (2 or 3) to be used
ldigits	GT number for SIM Server itself
NAME	Process name
CE	Hostname (CE means computing element)
EXEC	Program to be executed

authGateway.conf (Gateway routing configuration file)

Each line of this configuration file describes:

- The parameters used to form the destination SCCP address.
- The decimal value of the bearer service (bs) and/or teleservices (ts) indicating an authorization.
- A string associated with each bs or ts value. If no bs and no ts are specified, no authorization will be performed by the authGateway.

For each request, the first digits of the IMSI are compared with **odigits**. The first line of the configuration file that matches is selected for the current request. If the routing indicator (rri) specifies that the routing is done on GT, the leading digits are replaced with the new digits (ndigits) to perform the numbering plan translation. If a parameter is not present on the line, it will not be present in the SCCP destination address. If this is incompatible with the routing indicator, an error message will be issued when the MAP authentication Gateway is started.

After modifying this configuration file, the authGateway has to be restarted. Modification of the **ndigits** value may require the creation of a new GT rule (CREATE-GT command).

Routing based on GT (rri=0)

Numbering plan=E.214 (rnp=7)

odigits **24491** ndigits **35840** rri **0** rgti 4 rsn 6 rtt 0 rnp 7 rnai 4 bs 17:B11 bs 26:B1A ts 33:T21 odb 128:NotAuthorized ss 146:NotAuthorized

odigits **244911** ndigits **244911** rri **1** rpc **17** rsn **6** bs 17:B11 bs 26:B1A ts 33:T21 odb 128:NotAuthorized ss 146:NotAuthorized

odigits **244912** ndigits **244912** rri **1** rpc **18** rsn **6** bs 17:B11 bs 26:B1A ts 33:T21 odb 128:NotAuthorized ss 146:NotAuthorized

Table 56: Parameters Used in Create and Start Commands

Parameter	Description
rri	Routing indicator - 0 for GT (Global Title), 1 for PC/SSN (Point Code/Subsystem Number).
rgti	Local Global Title Indicator value. 4 for C7; 2 for A7. (Usually 4.)

Table 56: Parameters Used in Create and Start Commands (continued)

Parameter	Description
rssn	Subsystem Number of HLR.
rtt	(GT only) Translation Type (usually 0).
rnp	(GT only) Numbering Plan. 1 = E.164 = ISDN-TEL, 7 = E.214 = ISDN-MOB.
rmai	Nature of address indicator. (4 = INT)
bs	bs <i>dec:string</i> If the specified bearer service exists in HLR SIM-profile, then <i>string</i> is returned for further processing (see ProfileMap in simauth.aut/smsprov.aut).
ts	ts <i>dec:string</i> If the specified teleservice exists in HLR SIM-profile, then <i>string</i> is returned for further processing (see ProfileMap in simauth.aut/smsprov.aut).

ulcmmg.conf

```
LOCAL_HOST as1:2000
REMOTE_HOST as1:2001 [192.89.210.12]
```

Reserve port for authGateway application.

smsGateway Commands and Files

AS4StartSmsGw.mml — One Point Code/ Two SS7 Links

```
CREATE-PROCESS:NAME="SMS", CE="as1", EXEC="smsGateway -name SMS -port
2003 -host as1 -node MGW -conf conf/smsGateway.conf.100 -smsc 1935840123456
-appctx 3 -invktimeout 180";
```

```
START-PROCESS:NAME="SMS", CE="as1";
```

Table 57: Parameters Used in Create and Start Commands

Parameter	Description
name	Process name
port	Remote port specified in smsulcmmg.conf
host	Hostname
node	Node name (used in swmml command)
smsc	GT number of SMSC
appctx	MAP version (2 or 3) to be used
invktimeout	Timeout for sending SMS
NAME	Process name
CE	Hostname (CE means computing element)
EXEC	Program to be executed

smsGateway.conf

This configuration file describes the parameters used to form the destination SCCP address.

- One line containing **DestAddress** is used to address the destination node (HLR, MSC or SGSN). It is used only for SEND_ROUTING_INFO_FOR_SM in case of a Mobile Terminated SMS or to obtain the IMSI from the subscriber's MSISDN.
- The digits to address the HLR (that is, subscriber's MSISDN) are obtained from the client library.
- The digits to address the serving MSC (or SGSN) are obtained as a result of the SEND_ROUTING_INFO_FOR_SM message.
- If a parameter is not present on the line, it will not be present in the SCCP destination address. If this is incompatible with the routing indicator, an error message will be issued when the SMS Gateway is started.
- An optional parameter **rssn**, can be used to specify the destination ssn. This parameter will be used for routing only if the message is routed with a GT rule configured with SSN = DEF. If a normal GT is used (**SSN=value**), the value will overwrite the **rssn** defined in this file.
- All RoutingInfo messages are sent by default to SSN 6 (HLR).

The smsGateway has to be restarted after it is modified. Modification of the digits value may require the creation of a new GT rule (CREATE-GT command).

Example:

```
DestAddress rsn 8 rnai 4 rnp 1 rgti 4 rtt 0 rri 0
OrigAddress lpc 9619 lssn 7 ldigits 358402114300 oatype 1 oanp 1 oadigits
35840112233
ROCESS:NAME="SMS", CE="as1";
```

Table 58: Parameters Used in smsGateway.conf

Parameter	Description
rssn	Subsystem number of the MSC.
rnai	Nature of address indicator. 4 = INT
rnp	1 = E.164 = ISDN-TEL
rgti	(GT only) GTI value. 4 for C7; 2 for A7. (Usually 4.)
rtt	(GT only) Translation Type (usually 0)
rri	Remote routing indicator. 0 = GT, 1 = PC/SSN
lpc	Local (own) point code.
lssn	Local Subsystem Number (SSN)
ldigits	Local (own) GT number
oa*	Originating Address = SMS senders number
oatype	1 = international number
oanp	1 = E.164 = ISDN-TEL

Table 58: Parameters Used in smsGateway.conf (continued)

Parameter	Description
oadigits	Originating address digits

smsulcmmg.conf

```
LOCAL_HOST as1:2002  
REMOTE_HOST as1:2003 [192.89.210.12]
```

Reserve port for smsGateway application.

Appendix B

Reloading the Configuration with HUP Signals

Some configuration settings in Steel-Belted Radius are reloaded every time that Steel-Belted Radius receives a HUP (hang-up) signal. Other settings are loaded only when you stop and restart the Steel-Belted Radius server.

You can send a HUP signal by issuing the `sbrd hup` command.

You can also issue a HUP command manually. First, determine the process ID (PID) for the RADIUS service by issuing the `ps -ef | grep` command. Then, issue a `kill-HUP pid` command (where *pid* is the process ID for the radius service).

Table 59 identifies the files and file sections for which configuration settings are read when a HUP signal is received.

Table 59: Settings Reloaded with HUP Signals

File	Section	Parameter	Reloaded on HUP signal
cdracct.acc	[Bootstrap]	Enable	No
		InitializationString	No
		LibraryName	No
	[Settings]	CDRDirectory	Yes
		CdrDownlink	Yes
		CDRNodeID	Yes
		CdrType	Yes
		ConfigLog	No
		DefaultCUIDType	Yes
		SIMPartialCdrEnable	Yes
		SIMSessionCdrEnable	No
		SMSFixedFeeCdrEnable	No
		SMSPartialCdrEnable	No
		SMSSessionCdrEnable	No
		TimeThresholdEnable	Yes
		TimeThresholdSeconds	Yes
		UserPartialCdrEnable	No

Table 59: Settings Reloaded with HUP Signals (continued)

File	Section	Parameter	Reloaded on HUP signal	
gsmmap.gen		UserSessionCdrEnable	No	
		VolumeThresholdMegabytes	Yes	
		VolumeThresholdEnable	Yes	
	[Realms]	CatchAllRealm	No	
		NoRealm	No	
	[Settings]	ConfigLog	No	
	[<i>RealmName</i>]	AKA	No	
		Authorization	No	
		IMSI	No	
		MSISDN	No	
		SIM	No	
		SMS	No	
		[<i>targetmodule</i>] (where moduletype = GSM)	InitializationString	No
		LibraryName	No	
		ModuleType	No	
		RequestTimeoutMs	Yes	
		RequiredModuleVersionNumber	No	
		SymbolPrefix	No	
		[<i>targetmodule</i>] where moduletype = Database	DatabaseAccessorMethodName	Yes
		KeyForAuthorization	Yes	
	ModuleType	Yes		
eap.ini	[<i>authentication method</i>] such as [SQLAUTH]	EAP-Only	No	
		First-Handle-Via-Auto-EAP	No	
		EAP-Type	No	
		Available-EAP-Only-Values	No	
		Available-Auto-EAP-Values	No	
		Available-EAP-Types	No	
locspec.ctrl	[Bootstrap]	Enable	No	
		InitializationString	No	
		LibraryName	No	
	[Settings]	AttributeToIdentifyNAS	Yes	
		ConfigLog	No	
	[NAS-LIST]	NAS Designator	Yes	
	[<i>NAS Identifier</i>] section	GSM-Operator-Name	Yes	
		GSM-Location-Information	Yes	

Table 59: Settings Reloaded with HUP Signals (continued)

File	Section	Parameter	Reloaded on HUP signal
simauth.aut / simauth.eap	[Bootstrap]	GSM-Visited-Operator-Id	Yes
		GSM-Location-Name	Yes
		Enable	No
		InitializationString	No
		LibraryName	No
	[Settings]	AssignIpPoolByAttr	Yes
		AssignIpPoolDestAttr	
		ChargeableUserIdAttribute	Yes
		ChargeableUserIdInResponse	Yes
		ConfigLog	No
		EnableFailover	Yes
		EnableSIM	Yes
		EnableAKA	Yes
		FailoverTimeoutSec	Yes
		NumberOfTriplets	Yes
		ProfileIsUser	Yes
		PseudonymLifetimeDays	Yes
		PseudonymSecret	Yes
		ReauthenticationCountLimit	Yes
		ReauthenticationLifetimeSec	Yes
	ReauthenticationRealm	Yes	
	SenCUIDOnlyIfReceived	Yes	
	UseEAPResponseIdentity		
[Settings]	AssignIpPoolByAttr	Yes	
[IpPools]	<i>pool_name</i>	Yes	
[ProfileMap]	All settings	Yes	
smsauth.aut	[Bootstrap]	Enable	No
		InitializationString	No
		LibraryName	No
	[Settings]	ConfigLog	No
		NumberExpireSoonMessageThreads	Yes
		ProfileIsUser	Yes
		SendMsgBeforeExpiresSec	Yes
ValidateStationID	Yes		
smsmsg.gen	[Settings]	ConfigLog	No
		DefaultLanguageCode	Yes
		LanguageCodeSectionsList	Yes

Table 59: Settings Reloaded with HUP Signals (continued)

File	Section	Parameter	Reloaded on HUP signal
		SendEndTimeExpired	Yes
		SendEndTimeRemaining	Yes
		SendExpiringSoon	Yes
		SendPasswordReminder	Yes
		SendWelcome	Yes
	[Language]	EndTimeExpiredTemplate	Yes
		EndTimeRemainingTemplate	Yes
		ExpiringSoonTemplate	Yes
		PasswordMessageTemplate	Yes
		PasswordReminderTemplate	Yes
		WelcomeTemplate	Yes
smsprov.aut	[Bootstrap]	Enable	No
		InitializationString	No
		LibraryName	No
	[Settings]	ConfigLog	No
		DefaultAcctProvisionRequestSecs	Yes
		MaxAcctProvisionRequestSecs	Yes
		MaxPasswordResends	Yes
		MinAcctProvisionRequestSecs	Yes
		PasswordLength	Yes
		ProvisionedUnusedLifetimeSecs	Yes
		MinAcctProvisionRequestSecs	Yes
	[ProfileMap]	All settings	Yes
ss7ldapdb.gen	[Settings]	ConfigLog	No
		ExpiredAccountGracePeriodMin	Yes
		LDAPBaseDN	Yes
		LDAPMaxNumConnections	Yes
		LDAPPassword	Yes
		LDAPServerIPAddr	Yes
		LDAPServerPort	Yes
		LDAPServerTimeoutSec	Yes
		LDAPUserName	Yes
		LDAPVersion	Yes
		StaleAccountCleanerSweepFrequencyMin	Yes
		ExpiringSoonTemplate	Yes
		PasswordMessageTemplate	Yes
		PasswordReminderTemplate	Yes

Table 59: Settings Reloaded with HUP Signals (continued)

File	Section	Parameter	Reloaded on HUP signal
		WelcomeTemplate	Yes

Appendix C

SNMP Traps

The following SNMP traps have been added to the basic SNMP functionality in Steel-Belted Radius to support Steel-Belted Radius/SIM Server.

For information about how to configure SNMP settings in Steel-Belted Radius, refer to the *Steel-Belted Radius Reference Guide*.

Table 60: SNMP Traps

Trap Name	Steel-Belted Radius Trap Name	Comment
AUTHENTICATION SERVER FAILED TO INITIALIZE	funkSbrTrapServiceFailedInit	Treated as event.
MAP GATEWAY FAILED TO INITIALIZE	funkSbrTrapSS7MapGatewayFailedInit	Treated as event.
CDR GENERATION ERROR	funkSbrTrapSS7CDRGenerationError	Trap is sent once when Steel-Belted Radius/SIM Server fails to create CDR files.
	funkSbrTrapSS7CDRGenerationOK	Trap is sent when a CDR has been successfully written.
DATABASE ERROR	funkSbrTrapSS7AuthDatabaseError	Trap is sent once when Steel-Belted Radius fails to retrieve authorization information from any database.
	funkSbrTrapSS7AuthDatabaseOK	Trap is sent when Steel-Belted Radius successfully accesses the previously inaccessible database.
(none)	funkSbrTrapSS7ProvDatabaseError	Trap is sent once when Steel-Belted Radius/SIM Server fails to access the private LDAP database used for provisioning temporary subscriber accounts.
	funkSbrTrapSS7ProvDatabaseOK	Trap is sent when Steel-Belted Radius/SIM Server accesses the previously-inaccessible private LDAP database .
SS7 COMMUNICATIONS ERROR TO NSS	funkSbrTrapSS7CommunicationError	Trap is sent once when Steel-Belted Radius/SIM Server fails to communicate with an HLR or MSC.
	funkSbrTrapSS7CommunicationOk	Trap is sent when Steel-Belted Radius/SIM Server receives a response from a previously-inaccessible HLR or MSC.

Appendix D

Internal LDAP Directory

SIM Server uses an internal LDAP directory to perform two functions — extending the session information and storing temporary records used for Web-based authentication with SMS. Normally, there is no need to make any changes to this directory or its configuration file, `ss7ldapdb.gen`. Information about the configuration file is included here in case there is a unique application requirement compelling a change.

[Bootstrap] Section

The [Bootstrap] section of the `ss7ldapdb.gen` file contains the following settings:

```
[Bootstrap]
LibraryName=ss7ldapdb
Enable=1
```

Table 61: ss7ldapdb.gen [Bootstrap] Fields

Field	Description
LibraryName	Specifies the name of the library called when <code>gsmmap</code> runs. Default value is <code>ss7ldapdb</code> .
Enable	Set to 1 to enable the internal LDAP database. Set to 0 to disable the internal LDAP database. Default value is 1.

[Settings] Section

If you intend to use your own external LDAP server, you can configure the following items in the [Settings] section of `ss7ldapdb.gen`.

```
ConfigLog=ConsoleAndLog
LDAPServerIPAddr = 127.0.0.1
LDAPServerPort = 389
LDAPUserName = cn=Manager, o=sbrsms, c=US
LDAPPassword=password
LDAPBaseDN= o=sbrsms, c=US
LDAPVersion = 3
LDAPServerTimeOutSec = 0
LDAPMaxNumConnections = 300
StaleAccountCleanerSweepFrequencyMin = 30
ExpiredAccountGracePeriodMin = 7
```

Table 62: ss7ldapdb.gen [Settings] Fields

Field	Description
ConfigLog	<p>Specifies where LDAP configuration information will be logged. Options are:</p> <ul style="list-style-type: none"> ■ None – Do not log LDAP configuration information. ■ Log – Record LDAP configuration information in the Steel-Belted Radius log file. ■ Console – Display configuration information on the console only. ■ ConsoleAndLog (default) – Record LDAP configuration information in the Steel-Belted Radius log file and display configuration information on the Steel-Belted Radius console.
LDAPServerIPAddr	<p>The IP address of the LDAP server.</p> <p>The default is 127.0.0.1.</p>
LDAPServerPort	<p>The port number for the LDAP server.</p> <p>The default value is 389.</p>
LDAPUserName	<p>The string for the name of the LDAP server user account.</p> <p>The default is <code>cn=Manager, o=sbrsms, c=US</code>.</p>
LDAPPassword	<p>The LDAP server password. This password string must correspond to the credentials of the user account name in the previous field.</p> <p>The default is <code>password</code>.</p>
LDAPBaseDN	<p>The specification for the directory tree where LDAP SMS files, including provisioned accounts, are stored.</p> <p>The default is <code>o=sbrsms, c=US</code>.</p>
LDAPVersion	<p>The version of LDAP supported by <code>smsldapdb</code>. The version specified here must match the version number of the LDAP server. You can specify 2 or 3.</p> <p>The default is 3.</p>
LDAPServerTimeOutSec	<p>If a connection to the LDAP server cannot be made within the specified number of seconds, the transaction with the server is cancelled. A value of 0 means no transaction timeout.</p> <p>The default is 0.</p>
LDAPMaxNumConnections	<p>The maximum number of simultaneous connections with the LDAP server.</p> <p>The default is 300.</p>
StaleAccountCleanerSweepFrequencyMin	<p>Specifies how often accounts that have exceeded their grace period are purged from the LDAP database.</p> <p>The default is 30 minutes.</p>
ExpiredAccountGracePeriodMin	<p>Grace period, in minutes, within which an expired account may be re-billed, if the subscriber authenticates with the previously provided password during this period.</p> <p>The default is 7.</p>

Appendix E

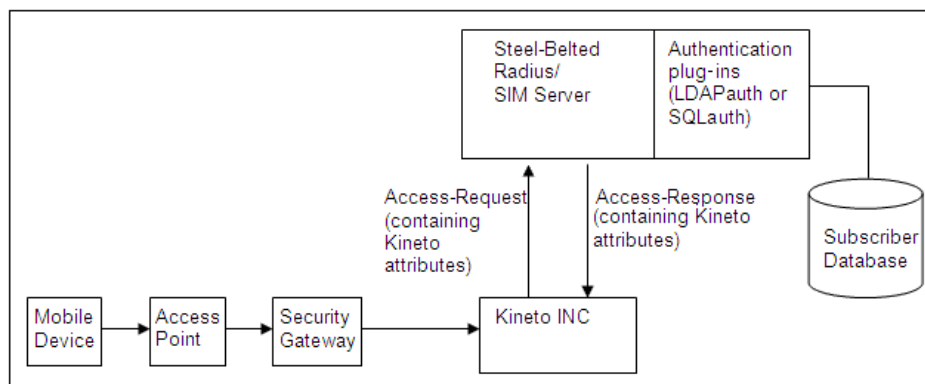
Kineto INC/SIM Server S1 Interface

The Kineto INC (IP Network Controller) is the component of the Kineto UMA network Controller (UNC) that manages subscriber access to voice and data mobile services. The Kineto INC-AAA (S1) interface Protocol Specification defines the interface requirements for communication between a AAA server and the Kineto INC.

Steel-Belted Radius/SIM Server is designed to comply with the requirements for a AAA server. This document explains how SIM Server interfaces with the Kineto INC and the tasks required for implementation.

Figure 29 illustrates the relationship between the Kineto INC and SIM Server.

Figure 29: Communication between Kineto INC and SIM Server



Attribute Handling Overview

The Kineto S1 Interface specification requires that certain Kineto Vendor Specific Attribute (VSAs) be sent with Access-Requests, Access-Accepts, and Access-Rejects as defined in the following publicly available Kineto documents:

- INC-AAA (S1) Interface Protocol Specification Version 1 (Revision 0.07)
- UMA Service Access Control January 2006

SIM Server is designed to comply with these specifications. Some of the Kineto VSAs contain compound attributes. SIM Server can process these attributes in an Access-Request by “flattening” them into single-payload attributes that can be more easily processed by Steel-Belted Radius and its associated plug-ins for authorization and authentication. In the returned Access-Accepts or Access-Rejects, SIM Server creates the compound Kineto VSAs from the flattened attributes before returning them to the Kineto INC device.

Table 63 on page 164, Table 64 on page 166, and Table 65 on page 168 list the Kineto VSAs that are converted to and from flattened attributes in Access-Requests, Access-Accepts, and Access-Rejects. All other Kineto attributes are passed between the Kineto INC and Steel-Belted Radius/SIM Server unchanged. If your processing requires handling of Kineto compound VSAs, use these conversion tables to identify the attributes to be passed between the Steel-Belted Radius AAA server and the Kineto INC.

Access-Request Conversion

Table 63 lists the Kineto Access-Request compound attributes that are flattened by SIM Server.

Table 63: Conversion from Kineto VSAs to Flattened Attributes

Kineto Attribute Name	Size in Octets	Format Description	Converted to Attributes
Kineto-UP-Client-Remote-Address	5 for IPv4 addresses and 17 for IPv6 addresses.	Consists of a discriminator byte followed by IPv or IPv6 address. IPv4: Discriminator octet has a value of 0x21 and the address is the following 4 octets. IPv6: Discriminator octet has a value of 0x57 and the address is the following 16 octets.	Converted to one of the following two attributes: Kineto-UP-Client-Remote-IPv4-Addr (type: ipaddr) or Kineto-UP-Client-Remote-IPv6-Addr (type: ipaddr)
Kineto-UMA-Classmark	2	Consists of multiple enumerated values within 2 octets.	Converted to all of the following four attributes: Kineto-UMA-Classmark-TURA (type: integer) Kineto-UMA-Classmark-UC (type:integer) Kineto-UMA-Classmark-GC (type:integer) Kineto-UMA-Classmark-RRS (type:integer)
Kineto-UMA-AP-Radio-Identity	7	Discriminator octet (always 0x1) followed by 6 octet MAC address.	Kineto-UMA-AP-Radio-MAC (type:string)

Table 63: Conversion from Kineto VSAs to Flattened Attributes (continued)

Kineto Attribute Name	Size in Octets	Format Description	Converted to Attributes
Kineto-UMA-MS-Radio-Identity	7	Discriminator octet (always 0x1) followed by 6 octet MAC address.	Kineto-UMA-MS-Radio-MAC (type:string)
Kineto-UMA-Location-Area-ID	5 if LAC is not present; 7 if LAC is present	<p>Contains MCC (Mobile Country Code), MNC (Mobile Network Code) and LAC (Location Area Code).</p> <p>If the LAC is not present in the original VSA sent in the Access-Request, the attribute will not be converted.</p> <p>The digits of MCC and MNC are encoded as BCD. MNC digit 3 may not be present, in which case its value will be 0xF.</p> <p>See Figure 30 on page 166 for an illustration of the encoding.</p>	<p>Kineto-UMA-Location-Area-MCC (type:string)</p> <p>Kineto-UMA-Location-Area-MNC (type:string)</p> <p>Kineto-UMA-Location-Area-LAC (type:string)</p>
Kineto-UMA-Cell-Identity	2	2-octet integer.	Kineto-UMA-Cell-Identity-Int4 (type:integer)
Kineto-UMA-AP-Service-Name	Number of octets in the string plus one	<p>Consists of an octet discriminator followed by a string value of the PAN or SSID.</p> <p>PAN: Discriminator octet has a value of 0x02 and the PAN is the following string.</p> <p>IPv6: Discriminator octet has a value of 0x01 and the SSID is the following string.</p>	<p>Kineto-UMA-AP-SSID (type:string)</p> <p>or</p> <p>Kineto-UMA-AP-PAN (type:string)</p>

Figure 30: Format of Kineto-UMA-Location-Area-Identification Attribute

bits	8	7	6	5	4	3	2	1	
	MCC digit 2				MCC digit 1				octet 1
	MNC digit 3				MCC digit 3				octet 2
	MNC digit 2				MNC digit 1				octet 3
	LAC								octet 4
	LAC (continued)								octet 5

Access-Accept Conversion

Table 64 describes the compound Kineto VSAs that are returned with Access-Accepts.

Table 64: Kineto attributes returned with the Access-Accepts

Flattened Attributes Converted From	Returned Kineto Attribute (Unflattened)	Format Description
Kineto-UMA-Service-Zone-Icon-Ind Kineto-UMA-Service-Zone-Name	Kineto-UMA-Service-Zone-Info	Consists of the Kineto-UMA-Service-Zone-Icon, followed by one octet containing the string length, followed by a string, extracted from Kineto-UMA-Service-Zone-Name.
Kineto-UMA-Location-Area-MCC Kineto-UMA-Location-Area-MNC Kineto-UMA-Location-Area-LAC	Kineto-UMA-Location-Area-ID	Encoded values of Kineto-UMA-Location-Area-MCC, Kineto-UMA-Location-Area-MNC, and Kineto-UMA-Location-Area-LAC as shown in Figure 30 on page 166.

Table 64: Kineto attributes returned with the Access-Accepts (continued)

Flattened Attributes Converted From	Returned Kineto Attribute (Unflattened)	Format Description
Kineto-UMA-GeogLoc-Latitude	Kineto-UMA-Geographical-Loc	Two geographical location types can be generated:
Kineto-UMA-GeogLoc-Longitude		<ul style="list-style-type: none"> ■ ellipsoid point (discriminator = 0x00, length = 7) for a latitude and longitude without any uncertainty.
Kineto-UMA-GeogLoc-Uncert-Circ (optional)		<ul style="list-style-type: none"> ■ ellipsoid point with uncertainty circle (discriminator = 0x10, length = 8) if Kineto-UMA-GeogLoc-Uncert-Circ is present.
		For both geographical location types, a latitude and longitude will be generated from Kineto-UMA-GeogLoc-Latitude and Kineto-UMA-GeogLoc-Longitude.
		The latitude and longitude will be encoded in complex ways [for more information, see <i>3GPP TS 23.032, Sections 5 (Shapes), 6 (Coding), and 7 (General message format and information element)</i>]. However, the converted latitude and longitude attribute formats must be encoded as ISO 6709 compliant string representations of decimal degrees <i>DD.DDDD</i> , for example 48.0234.
		Note: Any number of decimal places for the degrees will be accepted but the accuracy of the encoding depends on the format of the Kineto-UMA-Geographical-Loc attribute.
		Directions are expressed as follows:
		<ul style="list-style-type: none"> ■ northern latitudes—positive numbers ■ southern latitudes—negative numbers ■ east longitudes—positive numbers ■ west longitudes—negative numbers
		For the second geographical type, the expected format of the uncertainty circle is a string that contains a decimal number of meters.
		For a full description of the uncertainty encoding and its forward translation, see <i>3GPP TS 23.032, Section 6.2 (Uncertainty)</i> .

Access-Reject Conversion

Table 65 describes the Kineto compound VSA that is returned with an Access-Reject.

Table 65: Kineto attributes returned with the Access-Response

Flattened Attributes Converted From	Returned Kineto Attribute (Unflattened)	Format Description
Kineto-UMA-Service-Zone- Icon-Ind	Kineto-UMA-Service-Zone-Information	Consists of the Kineto-UMA-Service-Zone-I con-Ind, followed by one octet containing the string length, followed by a string, extracted from Kineto-UMA-Service-Zone- Name.
Kineto-UMA-Service-Zone- Name		

Configuring SIM Server for Kineto Attribute Handling

The following configuration activities are required to activate Kineto attribute handling:

- Configure the kinetoUMAAAttrHandler.ctrl file
- Configure the controlpoints.ini file
- Configure Steel-Belted Radius to recognize the Kineto attributes
- Develop applications for the S1 interface

Each of these configuration activities are described in the sections that follow.

Configuring the kinetoUMAAAttrHandler.ctrl File

The kinetoUMAAAttrHandler.ctrl file (located in the Radius directory) calls the appropriate library, enables use of the Kineto attribute handling features, and controls related settings.

To configure the kinetoUMAAAttrHandler.ctrl file:

1. Open the kinetoUMAAAttrHandler.ctrl file located in the Radius directory.
2. In the [Bootstrap] section of the kinetoUMAAAttrHandler.ctrl file, set `Enable=1`.
3. In the [Bootstrap] section of the kinetoUMAAAttrHandler.ctrl file, make sure the following lines exist and are not commented out:

```
LibraryName=kinetoUMAAAttrHandler.so
InitializationString= kinetoUMAAAttrHandler
```

4. In the [Settings] section of the kinetoUMAAAttrHandler.ctrl file, make sure the following line exists and is not commented out:

```
RemoveTranslatedAttributes=true
```

Example kinetoUMAAAttrHandler.ctrl file

```
[Bootstrap]
Enable=1
LibraryName=kinetoUMAAAttrHandler.so
InitializationString= kinetoUMAAAttrHandler
[Settings]
RemoveTranslatedAttributes=true
```

Table 66 explains the settings required in the kinetoUMAAAttrHandler.ctrl file to allow Kineto attribute handling.

Table 66: kinetoUMAAAttrHandler.ctrl Fields

Section	Field	Description
[Bootstrap]	LibraryName	Specifies the name of the library called. Set to kinetoUMAAAttrHandler.so
[Bootstrap]	Enable	Set to 1 to enable this file. Set to 0 to disable this file. Set to 1.
[Bootstrap]	InitializationString	Specifies the name of the initialization file for the library. Set to kinetoUMAAAttrHandler
[Settings]	RemoveTranslatedAttributes	Allows you to choose if the VSAs in an Access-Request which are translated by this plugin will be removed from the Access-Request. Set to true to delete the VSAs from the request. Set to false to leave the VSAs in the request.

Configuring the controlpoints.ini File

The controlpoints.ini file (located in the Radius directory) calls the attribute handler at the appropriate processing stages.

To configure the controlpoints.ini file:

1. Open the controlpoints.ini file located in the Radius directory.
2. Enter the following lines in the file:

```
[Auth-Initial-Request]
kinetoUMAAAttrHandler
[Auth-Final-Request]
kinetoUMAAAttrHandler
```

Table 67 explains the settings required in the controlpoints.ini file to allow Kineto attribute handling.

Table 67: controlpoint.ini File Settings

Field	Description
[Auth-Initial-Request] section	<p>Calls the attribute handler plug-in when the initial authorization request is received.</p> <p>Add the field:</p> <p>kinetoUMAAtrHandler</p>
[Auth-Final-Request] section	<p>Calls the attribute handler plug-in when authorization is complete.</p> <p>Add the field:</p> <p>kinetoUMAAtrHandler</p>

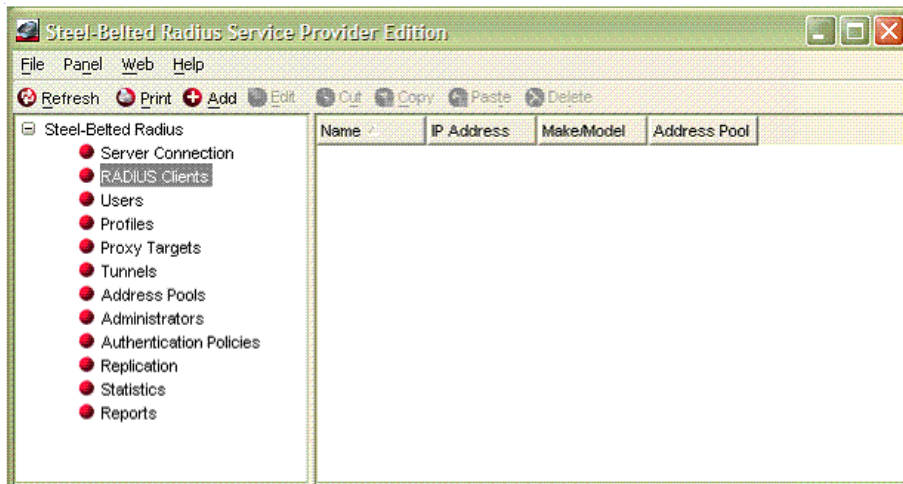
Configuring Steel-Belted Radius to Recognize the Kineto Attributes

You must configure Steel-Belted Radius to recognize the Kineto attributes by loading the Kineto dictionary file (.dct file).

To configure Steel-Belted Radius to recognize the Kineto attributes:

1. Run the Steel-Belted Radius Administrator and log into your Steel-Belted Radius server.
2. Click **RADIUS Clients**.

Figure 31: Main screen of the Steel-Belted Radius Administrator



3. Click **Add**.

The Add RADIUS Client dialog appears.

4. Select **Kineto S1** in the **Make/model** list and enter the details for your Kineto INC.



NOTE: Selection of **Kineto S1** in the **Make/model** list causes the Kineto dictionary file (.dct file) to be applied which includes the Kineto attributes.

Figure 32: Selection of Kineto S1 in the Add RADIUS Client dialog

Add RADIUS Client

Name: KINETO INC Any RADIUS Client

Description: The Kineto INC

IP Address: 10.10.100.2

Shared secret: ***** Unmask

Make/model: Kineto S1

Address pool:

EA Location Group:

Advanced

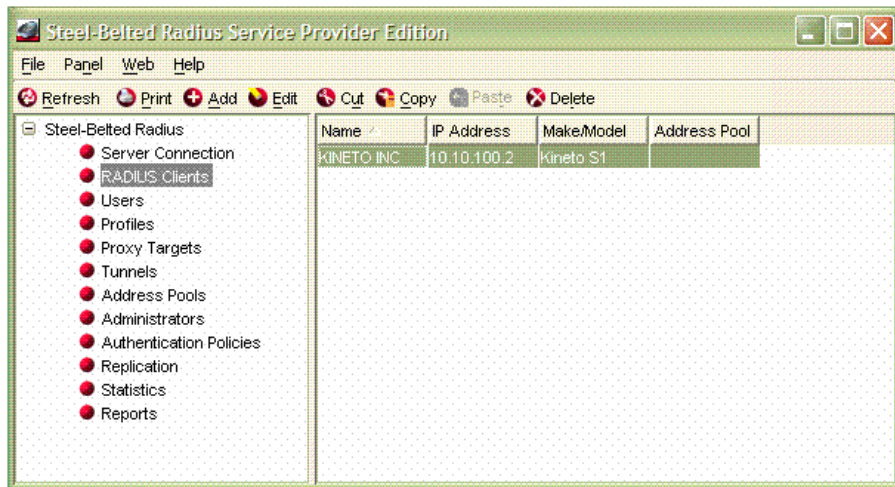
Use different shared secret for Accounting

Assume down if no keepalive packets after seconds

5. Click **OK**.

Kineto INC appears in your list of RADIUS clients.

Figure 33: Kineto added to the list of RADIUS clients

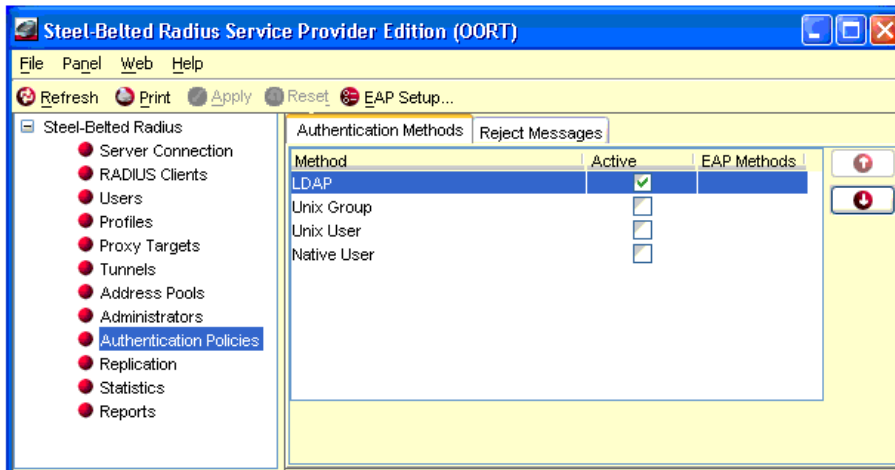


6. Click **Authentication Policies**.
7. Select **LDAP** or **SQL**.



NOTE: To cause LDAP or SQL to appear as a choice for Authentication Policies, set `Enable=1` in the `LDAPauth.aut` or `SQLauth.aut` configuration files.

Figure 34: LDAP Authentication Selected



Developing Applications for the S1 Interface

To implement the Kineto S1 interface with SIM Server, you must:

- Write your application using SQL stored procedures or LDAP scripting to conform with the requirements in the Kineto S1 interface specification.

- Configure and enable the LDAPauth plug-in or the SQLauth plug-in to authenticate subscribers using data stored in an LDAP directory or SQL database.

An example application using LDAP scripting and an LDAP directory schema is provided on the SIM Server CD in the Support_Files\SIM_Server\Kineto_S1_examples directory.

For more information about SQL stored procedures, LDAP scripting, the LDAPauth plug-in, and the SQLauth plug-in, see the *Steel-Belted Radius Administration Guide*, the *Steel-Belted Radius Reference Guide*, and the *Steel-Belted Radius LDAP Scripting Guide*.

Example Files

Example files are provided on the SIM Server CD (in the Support_Files\SIM_Server\Kineto_S1_examples directory) to demonstrate the implementation of the Kineto S1 interface. Table 68 describes each file.

Table 68: Example Files

File	Description
AAA_Kineto_S1_readme.txt	Provides an overview of the Kineto example files.
kinetoS1.schema	Schema for the example LDAP database.
ldapauth.aut	Configuration file for the LDAP plugin which authorizes subscribers using data stored in the LDAP directory defined by schema kinetoS1.schema. This file also includes a [Script] section which incorporates an LDAP scripting application designed to comply with the Kineto interface specification, while using the “flattened” and “unflattened” attributes defined in “Access-Request Conversion” on page 164, “Access-Accept Conversion” on page 166, and “Access-Reject Conversion” on page 168.

Appendix F

Glossary

Numerics

802.1X IEEE standard 802.1X. Standard for Local and Metropolitan Area Networks - Port-Based Network Access Control.

A

AAA Authentication, Authorization, and Accounting.

AC Access Controller.

accounting Recording of subscriber usage for billing and tracking purposes.

AKA Authentication and Key Agreement. AKA is an extension to the EAP protocol that enables authentication and session key distribution using a mechanism that is based on symmetric keys and that typically runs on a USIM.

AP Access Point.

AuC Authentication Center. The network element that provides the triplets for authenticating the subscriber.

authentication The process of verifying the identity of a wireless device and its user. This process is accomplished through transmission of identifying data at the time of connection.

authorization The process of specifying the permission (class of network services) granted to a subscriber.

B

BAOC Barring of All Outgoing Calls.

balun Balanced/unbalanced converter. A device used to match impedance between balanced and unbalanced lines, usually twisted-pair and coaxial cable.

BS Bearer Service. HLR authorization of service designation.

C

- CCB** Customer Care and Billing system.
- CDR** Call Detail Record. Call transaction record created by an MSC to track the network resources used by subscribers in making and receiving calls, so that billing systems can compute charges based on resource usage.
- CE** Computing Element.
- CG** Charging Gateway. Device that collects, validates, and consolidates CDRs from other network components for processing by the network billing system.

D

- D4** Framing format for T1 data transmission. A D4 frame consists of 192 data bits: 24 channels x 8 bits/channel plus a framing bit. D4 uses 12 consecutive frames to create a *superframe*.

E

- EAP** Extensible Authentication Protocol. The base protocol use for a variety of authentication methods with Radius and 802.1X.
- EAP-AKA** EAP method that allows authentication with a mobile subscriber USIM card.
- EAP-SIM** EAP method that allows authentication with a mobile subscriber SIM card.
- ESF** Extended Superframe Framing. A framing format for T1 data transmission that uses 24 consecutive frames to create a superframe.

F

- fixed-fee charge** Fee charged to user when the user acquires a service.
- FTP** File Transfer Protocol.

G

- GGSN** Gateway GPRS Support Node.
- GPRS** General Packet Radio Service. Packet-based wireless communication service for wireless phones and mobile computer users.
- GSM** Global System for Mobile Communications. A mobile telephone system that uses a SIM for subscriber identification.

H

- HLR** Home Location Register. The database of subscriber authentication information for a mobile network.
- hotspot** A WLAN Access Point offering network connectivity to the public.

I

- identity protection** Preventing an eavesdropper from discovering the identity of a user being authenticated.
- IMSI** International Mobile Subscriber Identifier. A unique subscriber identifier consisting of a three-digit Mobile Country Code (MCC), a two- or three-digit Mobile Network Code (MNC), and 10-digits-or-less Mobile Subscriber Identification Number (MSIN).
- ISO-639** International Standards Organization two-character letter code for the representation of names of languages.

L

- LDAP** Lightweight Directory Access Protocol.

M

- MAC (1)** Message Authentication Code.
- MAC (2)** Medium Access Control. Globally unique hardware address of a NIC or WLAN card.
- MAP** Mobile Application Part. The SS7 protocol standard that addresses registration of roaming users and the intersystem handoff procedure in wireless mobile telephony.
- MCC** Mobile Country Code. The MCC, together with the MNC, uniquely identify an operator and help identify the authentication center from which subscriber information should be retrieved.
- MNC** Mobile Network Code. The MNC, together with the MCC, uniquely identify an operator and help identify the authentication center from which subscriber information should be retrieved.
- MSC** Mobile Services Switching Center. Responsible for connecting calls together by switching packets from one network path to another. MSCs also provide information to support mobile service subscribers, including user registration, authentication, and location updating.
- MSISDN** Mobile Subscriber ISDN. Telephone number of mobile user.
- MTP** Message Transfer Part.

N

- NAI** Network Address Identifier.
- NAS** Network access server.
- NIC** Network interface card.
- nonce** Random value included in data exchanges to guarantee uniqueness and protect against replay attacks.
- numbering plan** The interpretation of the digits of an IMSI.

O

- ODB** Operator-Determined Barring. An HLR authorization of service designation that specifies that a subscriber is barred from service.

P

- permanent identity** The permanent identifier of a peer, including an NAI realm portion in environments where a realm is used. The permanent identity is usually based on the IMSI. Used on full authentication only.
- point code** The unique identifier for each node in an SS7 network.
- provisioning** The process of validating a user's telephone number, delivering a password to a user with SMS, and providing the user with a time-limited internet account.
- pseudonym identity** A pseudonym identifier of a peer, including an NAI realm portion in environments where a realm is used. Used on full authentication only.

Q

- quintets** The authentication data formed by the three UMTS values—RAND (random number), XRES (expected response), CK (cipher key), IK (integrity key), and AUTN (authentication token).

R

- RADIUS** Remote Access Dial-In User Service. Client/server technology that allows authentication of remote users in an IP network.
- re-authentication identity** The re-authentication identifier for a peer, including an NAI realm portion in environments where a realm is used. Used on re-authentication only.

S

- Signalware** The Ultricom SS7 protocol stack provided with Steel-Belted Radius/SIM Server.
- SIM** Subscriber Identity Module.
- SIM card** A SIM-based hardware SmartCard that contains the authentication keys for a GSM mobile telephone subscriber.
- SmartCard** A small card containing a computer chip that can store information, including authentication information and algorithms.
- SMS** Short Message Service. Protocol for sending messages of up to 160 alphanumeric characters between mobile subscribers and external systems such as email, paging, and voice mail.
- SS7** Signaling System 7. The network and protocols used to provide out-of-band signaling (control) for telephone services to support call establishment, billing, routing, and information exchange for the public switched telephone network.
- supplicant** WLAN client that must be authenticated before access to the network is granted.

T

- triplets** The authentication data formed by the three GSM values, RAND, Kc, and SRES.
- TS** Teleservice. HLR authorization of service designation.

U

- UMTS** Universal Mobile Telecommunications System. Type of mobile network (next generation after GSM) that uses the USIM card for authentication.
- USIM** UMTS Subscriber Identity Module.
- USIM card** A SIM-based hardware SmartCard that contains the authentication keys for a 3G mobile telephone subscriber.

V

- VSA** Vendor Specific (RADIUS) Attribute.
- VLR** Visitors Location Register.
- VPN** Virtual private network.

W

- Wi-Fi** Wireless local area network that uses the IEEE 802.11 a, b, or g radio protocols.

- W-CDR** Wireless LAN type of CDR.
- WISP** Wireless Internet Service Provider.
- WLAN** Wireless Local Area Network.

Index

Symbols

@realm tag..... 15

Numerics

3GPP-WLAN-APN-Id..... 93

A

AAA..... 2
access point name..... 92
Access-Accept with added attributes..... 96
AccessPointName/NASId..... 115
AccntProvisionRequestSecs..... 131
accounting..... 2
accounting options..... 106
Acct-Outbound-To-Proxy..... 90
activation target number..... 71, 79
adding attributes..... 96
address pools..... 95
AKA..... 154
AKA quintets..... 41, 63
ANSI SS7..... 2
APN. See access point name
appctx..... 50, 55
AsAddress..... 114
Asn1 CDR..... 110, 113, 122
AssignIpPoolByAttr..... 93, 155
AssignIpPoolDestAttr..... 93, 155
AttributeToIdentifyNAS..... 87
authentication..... 2
authenticator, helped..... 96
authGateway application..... 46
authGateway.conf..... 39, 49
AuthGatewayBarAccessIndicator..... 131
authorization..... 2, 175
 authGateway.conf..... 47
 disabling EAP-SIM..... 49
 EAP-AKA..... 14
 EAP-SIM..... 14
 HLR..... 49
 options..... 47, 48
 SIMAuth..... 11
Authorization parameter..... 154
Authorization string..... 41, 63
Auth-Outbound-To-Proxy..... 90

B

Bearer Service (BS)..... 48
binary version 1 CDR..... 110

binary version 2 CDR..... 110

C

CatchAllRealm..... 154
CauseForRecordClosing..... 116
CCITT/ITU SS7..... 2
CDR
 field formats..... 121
 fields..... 113
 filenames..... 110
 types..... 110
cdracct.acc..... 105, 129, 153
CDRDirectory..... 107, 153
CdrDownlink..... 109, 153
cdrdump..... 111
CDRNodeID..... 107, 153
CdrType..... 109, 110, 113, 153
ChangeTime..... 115
Chargeable User ID, see CUID
chargeable user id. See CUID
ChargeableUid..... 118
ChargeableUidLength..... 117
ChargeableUidType..... 117
ChargeableUserIdAttribute..... 124, 126, 155
Chargeable-User-Identity..... 124, 126
ChargeableUserIdInResponse..... 117, 126, 155
ChargingCharacteristics..... 116
ChargingId..... 114
ChargingType..... 116
ConcurrentTimeout..... 69
conf..... 50, 55
ConfigLog
 cdracct.acc..... 107, 153
 gsmmap.gen..... 60, 154
 locspec.ctrl..... 87
 simauth.aut..... 124, 155
 smsauth.aut..... 134, 155
 smsmsg.gen..... 136, 155
 smsprov.aut..... 130, 156
 ss7ldapdb.gen..... 156, 162
configuration script..... 17, 19
Connect..... 68
ConnectionType..... 116
ConnectTimeout..... 69, 77
conventions, text..... x
country code..... 88
CREATE-CPC..... 53
CREATE-GT..... 53
CREATE-PROCESS..... 50, 55

- CREATE-REMSSN 53
 CUID 117, 124, 126
- D**
- DatabaseAccessorMethodName 154
 DataVolumeDownlink 115
 DataVolumeUplink 115
 debug 50, 55
 DefaultAccntProvisionRequestSecs 156
 DefaultCUIDType 153
 DefaultLanguageCode 136, 155
 DependsOn 60, 135
 DestAddress 55
 dictionary file 39
 disabling EAP-SIM 49, 63
 DomainIndex 117
 dumpasn1 113
 Duration 115
- E**
- EAP helper 96
 eap.ini 98
 EAP-AKA 3, 11, 14
 EAP-SIM 3, 11
 EAP-SIM, disabling 49
 Enable
 cdracct.acc 106, 153
 gsmmap.gen 60
 ldapaccessor.gen 76
 locspec.ctrl 87
 simauth.aut 123, 124, 154, 155
 smsauth.aut 134, 155
 smsmsg.gen 135
 smsprov.aut 130, 156
 sqlaccessor.gen 68
 ss7ldapdb.gen 161
 EnableAKA 155
 EnableEAPAKA 124
 EnableEAPSIM 124
 EnableFailover 125, 155
 EnableSIM 155
 enabling failover 125
 EndTimeExpiredTemplate 156
 EndTimeRemainingTemplate 156
 ExChargingId 120
 ExChargingIdLength 120
 ExpiredAccountGracePeriodMin 156, 162
 ExpiringSoonTemplate 156
- F**
- failover 125
 FailoverTimeoutSec 155
 fast reauthentication 14, 15
 Framed-IP-Address 92, 96
 Framed-Ip-Address 93
- G**
- GgsnAddress 114
 global title 46
- GSM ix, 1
 GSM-Location-Information 88
 GSM-Location-Name 89
 gsmmap.gen 59, 129
 GSM-Operator-Name 88
 GSM-Visited-Operator-Id 88
- H**
- helped authenticator 96
 helper, EAP 96
 HLR 11, 41, 43, 46, 63, 125
 Home Locator Register, see HLR
 host 50, 55
- I**
- identity protection 14
 IMSI 154
 InitializationString
 cdracct.acc 106, 153
 gsmmap.gen 64, 65, 154
 locspec.ctrl 87
 simauth.aut 124, 154, 155
 smsauth.aut 134, 155
 smsprov.aut 130, 156
 installation
 Signalware 23
 SIM Server 17
 invkretry 50, 55
 invktimeout 50, 55
 IP address
 assignment 92
 pools 92, 95
 IpPools section 94
- K**
- key fields 82
 KeyForAuthorization 66, 82, 154
 Kineto INC 163
- L**
- LanguageCodeSectionsList 136, 155
 languages, for messages 135
 LDAP
 data accessor 76
 database 64, 66
 directory 3, 11, 67
 internal directory 161
 key fields 82
 server 161
 Steel-Belted Radius configuration 20
 ldapaccessor.gen 59, 60, 76
 LDAPBaseDN 156, 162
 LDAPMaxNumConnections 156, 162
 LDAPPassword 156, 162
 LDAPServerIPAddr 156, 162
 LDAPServerPort 156, 162
 LDAPServerTimeOutSec 162
 LDAPServerTimeoutSec 156
 LDAPUserName 156, 162

- LDAPVersion 156, 162
ldigits 55
lgti 50, 55
LibraryName 59, 153
 cdracct 106
 smsaut 134
 smsprov 68, 76, 130, 135, 161
license number 19
license, Signalware 24
link sets 43
links 43
lmsisdn 51, 55
lnai 51, 56
lnp 51, 56
LocalAddress 55
location attributes 88
location information 85
location name 89
LocationInfo 119
LocationName 118
locspectrl file 85
log entries 23
log thread 23
lpc 51, 53, 56
lri 51, 56
lssn 51, 53, 56
litt 51, 56
- M**
- M3UA 27
Man-Machine Language (MML) 42
MAP gateway 3
MAP protocol 52
max_requests 56
max_requests 51
max_RoutingInfo 56
MaxAccntProvisionRequestSecs 131, 156
MaxConcurrent 68, 77
MaxPasswordResends 131, 156
MaxWaitReconnect 69, 77
MethodName 77
MinAccntProvisionRequestSecs 131, 156
MML commands 42
MML settings, loading 58
Mobile Application Port protocol 52
ModuleType 154
monitor 51
MSC 41, 52
MSISDN 154
msisdn 47
MtAddress 115
- N**
- name 51, 56
NAS location 85
NasAddress 114
NasId 115
NAS-Identifier 85, 87
NAS-IP-Address 87
NAS-IP-Address) 85
NAS-List 88
NasPortType 114
NasTimeZone 114
ndigits 47
Network Address Identifier (NAI) 15
network equipment 63
no rst 51
node option 51, 56
NodeId 114
NoRealm 154
NumberExpireSoonMessageThreads 155
NumberOfTriplets 125, 155
- O**
- oadigits 54
oanp 54
oatype 54
ODB 48
odigits 48
Odyssey Client 11
OpenLDAP 22
operator name location identifier 88
Operator-Determined Barring (ODB) 48
OperatorName 120
- P**
- packages, Signalware 27, 33, 37
ParameterMarker 68
PasswordLength 131, 156
PasswordMessageTemplate 156
PasswordReminderTemplate 138, 156
PCI board, connecting 2
PdpAddress 120
PDPChargingCharacteristics 120
Perl script 37
permanent identity 14
plugins 86, 96
pools, IP address 95
port 51, 56
ProfileIsUser 134, 155
ProfileMap 127, 132
prot 51, 56
ProtocolType 115
ProvisionedUnusedLifetimeSecs 130, 156
provisioning
 configuring 130
 example 45
proxy
 requests 86
 targets 86
proxy.ini file 89
pseudonym identity 14
PseudonymLifetimeDays 125, 155
PseudonymSecret 125, 155
- Q**
- QoSTrafficClass 120
QueryTimeout 69, 77

R

RADIUS server	2
radius.ini	23, 129
radldapauth.aut	98
radsql.aut	98
radsqljdbc.aut	98
realm.pro file	xi, 90
realms	60
reauthentication	14, 15
identity	14
ReauthenticationCountLimit	125, 155
ReauthenticationLifetimeSec	125, 155
ReauthenticationRealm	125, 155
reboot	33
Record OpeningTime	115
Record SequenceNumber	116
RecordType	114
removing	
Signalware	33, 37
Signalware packages	37
SIM Server	36
RequestTimeoutMs	154
RequiredModuleVersionNumber	154
requirements, system	17
restart	33
rgti	48, 54, 57
rmai	48, 54, 57
rnp	48, 54, 57
route sets	43
routing, remote	47
routingInfoExp	56
rpc	48, 55
rri	48, 55
rssn	48, 55
rtt	48, 55, 57

S

S1 interface	163
SenCUIDOnlyIfReceived	155
SendCUIDOnlyIfReceived	124, 126
SendEndTimeExpired	156
SendEndTimeRemaining	156
SendExpiringSoon	136, 156
SendMsgBeforeExpiresSec	155
SendPasswordReminder	136, 156
SendWelcome	136, 156
ServedImsi	114
ServedMsIsdn	116
Server/name section	71
aut file	71, 79
Settings section	76
Signalware	2, 23
configuring	23
installation	23
MML commands	42
packages	33
starting	34
starting and stopping	34
stopping	34

SIGTRAN	23, 27, 31, 33, 41, 44, 45
SIM	154
SIM triplets	41, 63
SIMAuth	96
simauth	123
simauth.aut	93, 98, 123
simauth.eap	98
SIMPartialCdrEnable	108, 153
SIMSessionCdrEnable	108, 153
SMS	154
configuring options	133
interface requirements	139
message options	135
message templates	137
SMS gateway	
overview	3
SMS text message	41
SMS text messaging	63
smsauth	8
smsauth.aut	129, 133
smsc	57
SMSFixedFeeCdrEnable	107, 153
SMSGateway	52
smsGateway.conf	39, 129
smsmsg.gen	129
SMSPartialCdrEnable	108, 153
smsprov	
configuring provisioning	130
smsprov.aut	129
SMSSessionCdrEnable	108, 153
smsulcmmg.conf	39, 58
Solaris	1, 17, 35
SQL	68
SQL database	11
sqlaccessor.gen	60, 67
sqlaccessorjdbc.gen	67
SQLAuth	96
SS7	41
SS7 over IP	41
ss7ldapdb.gen	129, 156, 161
StaleAccountCleanerSweepFrequencyMin	156, 162
starting	
Signalware	33
SIM Server	22
starting Signalware	34
START-PROCESS	50, 55
stopping	
Signalware	34
SIM Server	22
stopping Signalware	34
stored procedures	67
supplicant	13
swmml	58
SymbolPrefix	154
system requirements	17

T

TeleService (TS)	48
TeliaSonera-Chargeable-UserId	126

text conventions x
 Timeout 77
 TimeThresholdEnable 109, 153
 TimeThresholdSeconds 108, 153
 tp_oa 57
 trace 51, 57
 tracefile 51, 57

U

ulcmmg.conf 39, 52
 Ulticom 2
 configuring 23
 uninstalling
 Signalware 37
 SIM Server 36
 upgrade procedure 35
 upgrading
 Signalware 38
 Solaris operating system 38
 UseEAPResponseIdentity 125, 155
 UserNameAndDomain 117
 UserPartialCdrEnable 107, 153
 UserSessionCdrEnable 107, 154
 USIM card 14

V

ValidateStationID 155
 VisitedOperatorID 119
 VolumeThresholdEnable 108, 154
 VolumeThresholdMegaBytes 108
 VolumeThresholdMegabytes 154

W

WaitReconnect 69, 77
 watchdog 22
 WelcomeTemplate 156, 157
 Wi-Fi Protected Access (WPA) 10
 WPA 10

