

STEEL-BELTED RADIUS™

Service Provider Edition

Administration Guide

UNIX and Windows Versions

Funk Software, Inc.
222 Third Street
Cambridge, MA 02142

617-497-6339
617-491-6503 (Technical Support)

© Copyright 1996-2003 Funk Software, Inc. All rights reserved.
5th Edition
July 2003

Steel-Belted Radius © 1996-2003 Funk Software, Inc. All rights reserved. Steel-Belted Radius is a registered trademark of Funk Software, Inc. This software contains material that is © 1994-1996 DUNDAS SOFTWARE LTD., all rights reserved. Portions Copyright 1993 Premia Corporation. Portions Copyright 1982-1995 Pervasive Software, Inc. All rights reserved. Microsoft, Windows, Windows NT, Windows 2000, Internet Explorer, and other Microsoft products referenced herein are either trademarks or registered trademarks of the Microsoft Corporation in the United States and other countries. Netscape and Netscape Navigator are registered trademarks of Netscape Communications Corporation in the United States and other countries. UNIX is a registered trademark in the U.S. and other countries, licensed exclusively through X/Open Company Limited. Sun, Sun Microsystems, Solaris, and all Sun-based trademarks and logos, Java, HotJava, JavaScript, the Java Coffee Cup Logo, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Raima, Raima Database Manager and Raima Object Manager are trademarks of Birdstep Technology.

Table of Contents

Preface

Audience	xvii
What's In This Manual	xvii
Typographical Conventions	xviii
Software-Level Text and Identifiers	xviii
User Interaction	xix
Files, Storage Devices, Websites	xix
Variable Text	xix
Related Documentation	xx
Technical Support	xx

Introduction

Related Products	3
System Requirements	3
Licensing	4

Installation

Installing and Starting under UNIX	6
Upgrading from a Previous Installation	10
Stopping and Starting the radius Daemon	12
Setting Up the Administrator Program	12
Installing under Windows	13
Before You Begin	13
Installing Steel-Belted Radius	14
Upgrading from a Previous Installation	15
Restoring a Previous Configuration	15
Starting and Stopping the RADIUS Service	17
Configuring for Authentication against Remote Domains	17
Upgrading from a 30-Day Trial Installation	18
Adding a License Key	19
Configuring the Server	19
UNIX-Specific Configuration	21
Configuring SecurID Authentication	24
Configuring TACACS+ Authentication	26

Oracle Software Changes.....	26
------------------------------	----

Concepts

RADIUS Basics	30
RADIUS Packets	32
RADIUS Configuration	32
Multiple RADIUS Servers.....	33
RADIUS Shared Secret	34
RADIUS Ports	35
Authentication.....	35
Authentication Methods.....	36
Configuring the Authentication Sequence.....	38
Configuring Authentication Methods	38
Advanced Options.....	41
Password Protocols	44
PAP	46
CHAP.....	46
MS-CHAP and MS-CHAP-V2	47
Accounting.....	47
Accounting Sequence	48
Accounting Spooling	51
Sessions List (Current Users Display).....	52
Attributes	52
Dictionaries	53
User Attribute Lists.....	54
Attribute Values	55
Default Values	56
Wildcard Support.....	57
Attribute Filtering	58
Profiles	58
Resolving profile and User Attributes	59
Request Routing.....	60
User-Names with a Single Delimiter	60
User-Names with Multiple Suffix Delimiters.....	62
User-Names with Multiple Prefix Delimiters.....	63
Request Routing by DNIS	64
Request Routing by Any Attribute	64
Local Services.....	65

Control Over Routing Methods	65
Proxy RADIUS	65
Proxy RADIUS Authentication	66
Proxy RADIUS Accounting	66
Proxy RADIUS Realms	66
Target Selection Within a Realm	68
Message-Authenticator Support	69
Proxy Fast-Fail	69
Static Proxy Accounting	70
Proxy AutoStop Feature	71
Tunnels	72
Tunnel Authentication Sequence	72
Configuring Tunnel Support	74
IP Address Assignment	75
Hints	76
Resource Management	76
Network Address Assignment	76
Concurrent Network Connections	79
Attribute Value Pooling	80
Phantom Records	80
Technical Bulletins	81

Administration

Administrator Program	84
Running the Administrator	84
Help with the Administrator	84
Exiting the Administrator	85
Servers Dialog	85
UNIX	86
Windows	86
RAS Clients Dialog	87
Adding a New RAS Client	88
Editing RAS Client Settings	88
Adding the <ANY> RAS Client	90
Removing a RAS Client	91
Users Dialog	91
Methods of Domain Authentication (Windows only)	91
Using the Users Dialog	95

Editing User Settings	96
Adding a Native User	100
Adding a Domain User or Domain Group (Windows only).....	102
Adding a Host User or Host Group (Windows only)	105
Adding a UNIX User or Group (UNIX only).....	107
Adding a SecurID User.....	107
Adding a TACACS+ User	109
Removing a User Entry.....	111
Profiles Dialog	111
Adding a Profile.....	112
Editing Profiles	112
Removing a Profile	112
Proxy Dialog	112
Adding a New Target.....	113
Editing Proxy Settings	114
Removing a Target.....	116
Steel-Belted Radius as a Target.....	116
Proxy RADIUS as an Authentication Method.....	118
Tunnels Dialog.....	118
Adding a Tunnel	119
Editing a Tunnel.....	120
Removing a Tunnel.....	120
IP Pools Dialog	121
Adding an IP Address Pool.....	121
Editing an IP Address Pool.....	122
Removing an IP Address Pool	122
Specifying IP Address Assignment in User/Profile Records.....	122
NAS-Specific IP Address Pools	123
Specifying IP Address Assignment from a DHCP Server.....	124
IPX Pools Dialog	128
Adding an IPX Pool	129
Editing an IPX Pool	129
Removing an IPX Pool	129
Specifying Pooled IPX Network Numbers in User/Profile Records ..	130
Access Dialog	130
UNIX Only	130
Windows Only	132
Configuration Dialog	134
Reject Messages.....	135

Tunnel Name Parsing.....	136
Authentication Methods Configuration	137
Log Files	137
Import/Export Capabilities	138
Exporting to a RADIUS Information File	138
Importing from a RADIUS Information File.....	139
Importing from Other File Formats	141

Logging, Monitoring, and Reporting

A Window on Operations	144
Radius Log File.....	144
Level of Logging Detail.....	145
Authentication Log File	146
Authentication Log File Format	146
First Line Headings.....	147
Comma Placeholders	147
Accounting Log File	148
Accounting Log File Format.....	149
First Line Headings.....	150
Comma Placeholders	150
Standard RADIUS Accounting Attributes.....	151
Statistics Dialog	152
Authentication Statistics	153
Accounting Statistics	154
Proxy Statistics	157
Resetting Server Statistics	158
Sessions List	158
Modifying the Sessions List Columns.....	160
Sorting the Sessions List (Windows only).....	160
Refreshing the Sessions List	160
Deleting Entries from the Sessions List.....	160
Reporting Capabilities	161
Setting Report Options (Windows only)	161
Creating a Report	161
Windows NT Performance Monitor	163
Windows NT Events	168
Informational Events.....	169
Warning Events.....	170

Error Events	172
--------------------	-----

Server Configuration

Server Configuration Files	176
access.ini File	177
account.ini File	178
account.ini [Alias/name] Sections	178
account.ini [Attributes] Section	179
account.ini [Configuration] Section	180
account.ini [Settings] Section	181
account.ini [TypeNames] Section	183
admin.ini File	184
authlog.ini File	186
authlog.ini [Alias/name] Sections	186
authlog.ini [Attributes] Section	187
authlog.ini [Configuration] Section	188
authlog.ini [Settings] Section	188
blacklist.ini File	191
bounce.ini File (Windows only)	192
classmap.ini File	193
classmap.ini [AttributeName] Section	193
dhcp.ini File	194
dhcp.ini [Settings] Section	195
dhcp.ini [Pools] Section	196
pool.dhc Files	196
pool.dhc [Settings] Section	197
pool.dhc [Request] Section	197
pool.dhc [Reply] Section	199
Reconfiguring Pools	200
events.ini File	200
events.ini [EventDilutions] Section	200
events.ini [Suppress] Section	201
events.ini [Thresholds] Section	201
filter.ini File	202
Filter Rules	203
Order of Filter Rules	204
Values in Filter Rules	205
Referencing Attribute Filters	206

lockout.ini File	207
Clearing Locked-Out Accounts	207
radius.ini File	208
radius.ini [Addresses] Section	208
radius.ini [Certificate] Section	209
radius.ini [AuthRejectLog] Section	210
radius.ini [Configuration] Section	212
radius.ini [CurrentSessions] Section	216
radius.ini [FailedAuthOriginStats] Section (Windows only)	216
radius.ini [IPPoolSuffixes] Section	217
radius.ini [LDAP] Section	218
radius.ini [LDAPAddresses] Section	218
radius.ini [NTDomain] Section (Windows only)	218
radius.ini [Ports] Section	220
radius.ini [SecurID] Section	221
radius.ini [SecurID] Section - Enhanced Token Caching	222
radius.ini [Self] Section	224
radius.ini [StaticAcctProxy] Section	224
radius.ini [Strip] Section	224
radius.ini [ValidateAuth] and [ValidateAcct] Sections	226
redirect.ini File	228
redirect.ini [Settings] Section	228
redirect.ini [ClientExclusionList] Section	229
spi.ini File	229
spi.ini [Keys] Section	230
spi.ini [Hosts] Section	230
tacplus.ini File	231
tacplus.ini [ServerInfo] Section	231
update.ini File	232
update.ini [HUP] and [USR2] Sections	232
Sample update.ini File	234
vendor.ini File	235
vendor.ini [Vendor-Product Identification] Section	235
vendor.ini Product-Scan Settings	236
Dictionary Files	239
Windows	239
UNIX	239
Editing Dictionary Files	240
Include Records	240

ATTRIBUTE Records	242
MACRO Records	245
OPTION Records	246
services File	246
Attribute Value Pools (*.rr files)	247
Auto-Restart Files (UNIX only)	249
Perl SNMP Support	249
S90radius Script	250
radiusd Script	250

Realm Configuration

Stage One of Realm Configuration	256
Realm Configuration Files	257
Configuring a Proxy RADIUS Realm	257
Configuring a Directed Realm	263
radius.ini Realm Settings	268
proxy.ini File	268
proxy.ini [AttributeMap] Sections	269
proxy.ini [Configuration] Section	272
proxy.ini [Directed] Section	272
proxy.ini [DirectedAcctMethods] Section	273
proxy.ini [Interfaces] Section	274
proxy.ini [Processing] Section	275
proxy.ini [Realms] Section	275
proxy.ini [StaticAcct] Section	276
Proxyrl.ini	277
Proxy RADIUS Configuration (.pro) File	278
Proxy RADIUS [Auth] Section	279
Proxy RADIUS [Acct] Section	281
Proxy RADIUS [AutoStop] Section	284
Proxy RADIUS [Called-Station-ID] Section	285
Proxy RADIUS Target Selection Rules	286
Proxy RADIUS [FastFail] Section	289
Proxy RADIUS [ModifyUser] Section	290
Proxy RADIUS [SpooledAccounting] Section	290
Directed Realm Configuration (.dir) File	292
Directed Realm [Auth] Section	293
Directed Realm [AuthMethods] Section	294

Directed Realm [Acct] Section	295
Directed Realm [AcctMethods] Section	296
Directed Realm [Called-Station-ID] Section	297
Directed Realm [ModifyUser] Section	297

Extensible Authentication Protocol

EAP Concepts	300
Handling EAP Requests	300
Automatic EAP Helpers	301
Authentication Request Routing	302
EAP-NAK Notifications	304
Reauthenticating Connections	304
EAP Types	305
EAP-PEAP	305
LEAP	305
EAP Generic-Token	306
EAP MD5-Challenge	306
eap.ini File	306
Configuring For EAP-TTLS and EAP-PEAP	308
ttsauth.aut and peapauth.aut files	308
Examples of EAP Usage	314
LEAP	315
LEAP and EAP MD5-Challenge	316
EAP-TTLS and EAP-TLS	317
EAP-PEAP	320

SNMP

Introduction	324
SNMP Management Information Base (MIB)	324
SNMP Manager and SNMP Agent	325
SNMP Sub Agent	325
Steel-Belted Radius SNMP Sub Agent	325
SNMP Traps and Alarms	326
Dilution and Threshold	327
Acting on Information	328
SNMP Access Control	328
Counter Statistics	328
RADIUS-Authentication-Client Statistics	328

RADIUS-Authentication-Server Statistics	329
RADIUS-Accounting-Client Statistics	329
RADIUS-Accounting-Server Statistics	329
Rate Statistics.....	330

LDAP Configuration Interface

LDAP Configuration Interface	332
LDAP Command Line Utilities	332
LDAP Version Compliance	333
LDAP TCP Port	334
LDAP Virtual Schema	335
Bind Request.....	339
Uppercase and Lowercase.....	339
Attributes	340
IP Addresses	340
IPX Addresses.....	340
Substrings.....	340
Hexadecimal Values	340
Password Syntax	341
Profiles, Check-Lists, and Return-Lists.....	341
LDAP Command Examples.....	342
Searching for Records.....	342
Modifying Records	343
Adding Records	346
Deleting Records.....	347
LDIF File Examples.....	348
Adding RADIUS Clients with LDIF	349
Adding Users with LDIF	350
Adding Proxy Targets with LDIF	353
Adding Tunnels with LDIF.....	354
Adding IP Address Pools with LDIF	355
Adding IPX Address Pools with LDIF	355
Configuring a RADIUS Server with LDIF	356
Statistics Variables (LCI Only).....	357
Counter Statistics	357
Rate Statistics.....	359

SQL Authentication

SQL Authentication	364
SQL Authentication Process	365
Configuring SQL Authentication	365
Using the SQL Authentication Header File	366
Using Multiple SQL Databases	366
Connecting to the SQL Database	367
SQL Statement Construction	368
Password Parameters	369
Overlapped Execution of SQL Statements	370
The %Result Parameter	371
SQL Authentication and Password Format	372
SQL Authentication Header (.aut) File	373
SQL Authentication [Bootstrap] Section	373
SQL Authentication [FailedSuccessResultAttributes] Section	374
SQL Authentication [Failure] Section	374
SQL Authentication [Results] Section	375
SQL Authentication [Server] Section	378
SQL Authentication [Server/name] Sections	379
SQL Authentication [Settings] Section	380
SQL Authentication [Strip] Sections	383
Working with Stored Procedures in Oracle	385

SQL Accounting

SQL Accounting	388
Configuring SQL Accounting	389
Using the SQL Accounting Header File	389
Using Multiple SQL Databases	389
Connecting to the SQL Database	390
SQL Statement Construction	390
INSERT Statement and VALUES Section	391
Using Multiple SQL Statements	394
Overlapped Execution of SQL Statements	394
SQL Accounting Return Values	395
SQL Accounting Header (.acc) File	395
SQL Accounting [Bootstrap] Section	396
SQL Accounting [Settings] Section	396
SQL Accounting [Type] Sections	397

Working With Stored Procedures	400
Load Balancing Example	401

LDAP Authentication

External LDAP Authentication	404
LDAP Variable Table	405
Types of LDAP Authentication	405
Configuring LDAP Authentication	407
LDAP Database Schema	408
LDAP Authentication and Password Format	410
LDAP Authentication Header (.aut) File	412
LDAP Authentication Variable Names	412
LDAP Authentication [Response] Section	412
LDAP Authentication [Attributes/name] Sections	415
LDAP Authentication [Search/name] Sections	416
LDAP Authentication [Request] Section	419
LDAP Authentication [Defaults] Section	421
LDAP Authentication [Server/name] Sections	421
LDAP Authentication [Server] Section	424
LDAP Authentication [Settings] Section	425
LDAP Authentication [Failure] Section	428
LDAP Authentication [Bootstrap] Section	429
LDAP Authentication Sequence	430
LDAP Authentication Examples	431
Bind Authentication with Default Profile	431
BindName Authentication with Callback Number Returned	433
LDAP Bind with Profile Based on NAS Device	434

Quick Reference

When to Stop and Restart the Server	436
Configuration Files by Feature	438
Configuration Files by Name and Extension	442
Steel-Belted Radius Vendor-Specific Attributes	446

Technical Bulletins

LDAP Support for Novell NDS	448
Service Type Mapping	452

Configuration	452
servtype.ini File	454
User Name Transform	459
Operation	459
Routed Proxy	461
Operation	461
CCA Support for 3COM	463
Configuration	463
Setting User and Profile Attributes	463
Ascend Filter Translation	465
Configuration	465
Syntax	465
ldapauth Extensions	467
GlobalProfile Attribute	467
ProfileData Attribute	468
Modifying ldapauth.aut	468
Ericsson's e-h235 Authentication Protocol	470
Operation	470
Configuration	470
Uniport Plug-In	471
Operation	471
Configuration	471

Index

Preface

This manual describes how to install, configure, and administer Steel-Belted Radius for UNIX and Windows NT/Windows 2000/Windows XP.

Audience

This manual is intended for network administrators responsible for implementing and maintaining authentication, authorization, and accounting services for an enterprise. This manual assumes that you are familiar with general RADIUS and networking concepts and the specific environment in which you are installing Steel-Belted Radius.

What's In This Manual

This manual is organized as follows:

- Chapter 1, “Introduction,” presents an overview of Steel-Belted Radius and describes installation and licensing requirements for Steel-Belted Radius.
- Chapter 2, “Installation,” describes how to install Steel-Belted Radius on a UNIX or Windows computer and how to add a Steel-Belted Radius license key.
- Chapter 3, “Concepts,” summarizes important concepts relating to operation of Steel-Belted Radius.
- Chapter 4, “Administration,” describes how to use the Administrator program to configure Steel-Belted Radius.
- Chapter 5, “Logging, Monitoring, and Reporting,” describes how to use the logging and monitoring facilities in Steel-Belted Radius.
- Chapter 6, “Server Configuration,” describes the initialization and configuration files used by Steel-Belted Radius.
- Chapter 7, “Realm Configuration,” describes how to configure and maintain directed and Proxy RADIUS realms.

- Chapter 8, “Extensible Authentication Protocol,” presents an overview of EAP types and describes how to configure EAP in Steel-Belted Radius.
- Chapter 9, “SNMP,” presents an overview of SNMP components and describes the statistics available for Steel-Belted Radius servers and clients.
- Chapter 10, “LDAP Configuration Interface,” describes how to use public domain LDAP utilities to populate a Steel-Belted Radius server database.
- Chapter 11, “SQL Authentication,” describes how to configure authentication against records stored in an external SQL database.
- Chapter 12, “SQL Accounting,” describes how to configure Steel-Belted Radius to write accounting information to an external SQL database.
- Chapter 13, “LDAP Authentication,” describes how to configure authentication against records stored in an external LDAP database.
- Appendix A, “Quick Reference,” provides a summary of critical operational information and lists configuration files by feature and by name.
- Appendix B, “Technical Bulletins,” presents technical tips for configuring Steel-Belted Radius to interoperate with equipment and facilities offered by other vendors.

An index for this manual appears in the back of this book.

Typographical Conventions

To make this documentation easy to read, and the text as unambiguous as possible, the following conventions have been adopted.

Software-Level Text and Identifiers

Software-level text and other identifiers (attribute names, values, etc.) appear in this manual in a plain monospace font, as below. This font is also used for displaying the contents of computer files, text visible on status messages, and other sorts of technical details.

Consider the following example:

```
[EventDilutions]  
SQLConnectFailure=8
```

User Interaction

Text that guides your interaction with the software's user interface appear in another font. This is used to specify particular keys on the keyboard (such as **[Esc]**), a string of text (such as “enter **YES**”), programs to invoke from the command line (such as “Now run the installation program by entering **installme** from the command line”), particular buttons or components of the user interface (such as “select the **OK** button” or “click on the **Reboot on disconnect** checkbox”).

Menu commands are presented as the name of the menu, followed by the > sign, and concluding with the name of the command itself. For example, the **Cut** menu command on the **Edit** menu would be written as **Edit > Cut**. If the item on the menu is not a command but a hierarchical menu, the menu chain is longer. For example, if the **Edit** menu has an entry called **Paste As...** which leads to a hierarchical menu which contains a command called **Text**, this chain of items would be written as **Edit > Paste As... > Text**.

Files, Storage Devices, Websites

All files, storage devices, and web-sites appear in the text shown in the following examples:

For more information, go to www.tellmemore.com

Now copy **SPACEHOG.DAT** to your C: drive.

Variable Text

Variable text, where you replace a placeholder with your own information, appears in *italics*. Examples of variables include names, dates, and user selections. For example, to demonstrate that you should enter your name and password when prompted to do so, the interaction would be presented as follows:

Enter your name: ***YourName***

Password: ***YourPassword***

File names and computer text can also be displayed in italics to indicate that the exact text can change and that it is up to you to supply it. For example, the section of the configuration file shown above might be explained as:

```
[EventDilutions]
EventName=DilutionCount
EventName=DilutionCount
...
```

where *EventName* is the name of a known event and *DilutionCount* is an integer count.

Related Documentation

Most of the information in this manual is available in the online help available from the Administrator program.

You can consult our online **Vendor Information** file for information about using Steel-Belted Radius with many popular brands of Remote Access Server and Firewall. You can access this file by starting the Administrator, choosing the RAS Clients dialog, and clicking the **Vendor Info** button. (In Windows, you can also access the file by selecting **Help > Vendor Info** from the **Administrator** menu bar.) For more information about configuring your access servers and firewalls, consult the manufacturer's documentation provided with each device.

Please also review the `readme.txt` file, which contains late-breaking information not available in this manual.

Technical Support

If you have any problems installing or using Steel-Belted Radius, there are various resources available to help you.

This manual and the `readme.txt` files provided with the product may contain the information you need to solve the problem you are having. Please re-read the relevant sections. You may find a solution you overlooked.

A range of support options is also available. Refer to the enclosed brochure for information about the support plan that best meets your needs.

If you haven't already done so, please fill out and return the enclosed Registration Card to ensure that you are notified of upgrades and of new networking products as they become available.

Introduction

1

- Related Products
- System Requirements
- Licensing

Thank you for selecting Steel-Belted Radius.

Steel-Belted Radius is a complete implementation of the widely-used IETF standards-track RADIUS (Remote Authentication Dial-In User Service) protocols. It interfaces with a wide variety of network access equipment, and authenticates remote and WLAN users against numerous back-end databases — allowing you to easily consolidate the administration of all your remote and WLAN users, however they connect to your network.

Steel-Belted Radius delivers a total RADIUS solution on the scale required by Internet Service Providers and carriers. It provides the power and flexibility you need to manage the delivery of enhanced services to your customers. And, it integrates with all aspects of your NOC, from customer authentication and service delivery to your back-office accounting and billing system.

Highlights of Steel-Belted Radius include:

- Advanced proxy features let you easily authenticate users against RADIUS servers at other sites.
 - You have a choice of user name format, and you can configure routing based on user name decoration, DNIS, or specific attributes.
 - You can selectively modify attributes as proxy packets flow to and from Steel-Belted Radius.
 - You can specify groups of proxy target servers that handle proxy requests according to load-balancing or retry strategies — for the best performance and reliability.
- Directed authentication and accounting features simplify the hosting of RADIUS services by allowing Steel-Belted Radius to provide services uniquely for each of your customers. Incoming requests can be directed to specific authentication or accounting methods based on user name decoration or DNIS.
- Advanced external authentication features let you authenticate against multiple, redundant SQL or LDAP databases according to configurable load balancing and retry strategies, ensuring the highest level of service delivery to your users.
- You can control the time periods during which each user is allowed access. An access request is granted only during a user's allowed access hours; otherwise it is refused, even if the user presents valid credentials.
- Your choice of interface lets you configure Steel-Belted Radius via a graphical Administrator program, or via LDAP (either programmatically or at the command line prompt).

- Administrative access levels can be defined and applied to user or group accounts on the server machine. Read, write, and read/write access can be applied selectively to various categories of configuration data: Users, RAS Clients, Proxies, Statistics, and so on.
- Auto-restart permits the Steel-Belted Radius server to restart itself automatically if it experiences a shutdown.
- **UNIX only:** SNMP support lets you centrally monitor Steel-Belted Radius from your SNMP console, in the same manner as you monitor other devices and services on your network. Steel-Belted Radius offers full SNMP support including SNMP traps and alarms.
- **Windows only:** Perfmon counter and NT event support let you centrally monitor Steel-Belted Radius using Windows NT platform tools, in the same manner as you monitor other services on your network.

Related Products

Contact Funk Software for information about the Add-On Policy Servers that are available for Steel-Belted Radius. These Policy Servers provide specialized functionality such as the management of concurrent access limits across multiple copies of Steel-Belted Radius.

System Requirements

UNIX

The Steel-Belted Radius for UNIX software package includes the server daemon, a Java-based administration user interface, and various dictionary and database files to support authentication.

The Steel-Belted Radius for UNIX software requires Solaris 2.6, 7, 8, or 9 (or later) on a SPARC workstation or server with at least 64 megabytes of working memory. Installing Steel-Belted Radius requires 105 megabytes of space on the hard disk; hard disk requirements for running Steel-Belted Radius depend on your system's product configuration.

The administration UI requires a Java-capable browser that understands signed Java applets. Browsers that meet these criteria include Netscape Navigator 4.08 or later

on Windows NT/2000 and Solaris, and Internet Explorer 4.0 or later on Windows 95 and NT.

Steel-Belted Radius supports Oracle WorkGroup Server v7.3.3 and later for use as an external database for RADIUS authentication and accounting.

Windows

Steel-Belted Radius for Windows runs on a Windows workstation or server using one of the following operating systems:

- Windows NT 4.0 with Service Pack 6
- Windows 2000 (all editions)
- Windows XP (all editions)
- Windows Server 2003 (all editions)

TCP/IP must be configured on the Windows host for Steel-Belted Radius to function properly.

Steel-Belted Radius can be administered from the local Windows machine on which it is running, or it can be administered remotely from another Windows machine.

Steel-Belted Radius is compatible with any SQL database server that is ODBC compliant. This includes servers made by Oracle, Sybase, and Microsoft.

Licensing

Steel-Belted Radius can be installed on a single workstation or server.

For details about licensing, please refer to the enclosed license agreement or contact Funk Software directly.

Note: The Steel-Belted Radius license permits you to configure a total of 10 directed authentication and/or directed accounting methods. If you need additional methods, contact Funk Software to purchase blocks of additional licenses.

Installation

2

- Installing and Starting under UNIX
- Installing under Windows
- Upgrading from a 30-Day Trial Installation
- Adding a License Key
- Configuring the Server

Installing and Starting under UNIX

This section describes how to install the Steel-Belted Radius software onto a UNIX server or workstation.

Note: *SNMP support requires that you install the Solstice Enterprise Agent (SEA) from Sun Microsystems. You can install SEA before or after you install Steel-Belted Radius. To download a free copy of SEA and configure it to work with Steel-Belted Radius, see “Configuring SNMP Support” on page 22.*

To install and start Steel-Belted Radius under UNIX:

- 1 Review the system requirements and list of documents in “System Requirements” on page 3.
- 2 Copy the installation files to the UNIX machine. Set your working directory to the directory to which you’ve copied the files.
- 3 To display detailed information about the Steel-Belted Radius installation script and its options, type the following command:
sh install.sh -info
- 4 Run the install.sh script with the **-all** option:
sh install.sh -all
- 5 When the script prompts you for the directory where you want to install the Steel-Belted Radius software (the *server directory*), enter a full pathname.
If the directory does not already exist, the script creates it.
- 6 The script prompts you to enter a license key. If you have purchased the product, you’ll find this number on a sticker affixed to the license agreement in your product package.
 - If you type **y** and press **[Enter]**, you are prompted to enter the license key. Type the number and press **[Enter]**. The script creates the license file and copy it to the server directory.
 - If you type **n** and press **[Enter]**, the software defaults to 30-day evaluation mode, allowing use of the product’s full feature set for a limited period. When a list of products available for trial installation appears, enter the number matching the product you want to evaluate and press **[Enter]**.
- 7 If you have installed a previous version of this same edition of Steel-Belted Radius on this computer, during the next several steps the install.sh script can detect the following items on the machine:

- A radius daemon that is already running
- Steel-Belted Radius configuration files
- Steel-Belted Radius database files

The script checks for these items before copying any files to the server directory. If you have never installed Steel-Belted Radius on this UNIX machine before, skip to step 13.

- 8 The `install.sh` script checks for a running server:

```
Checking for a running server
```

If the script detects a running radius daemon, the following prompt is displayed; if not, you can skip to step 10:

```
Server is running with pid x
Stop radius server and unconfig/uninstall before
installing new version
```

- 9 If you see the above prompt, complete the following steps before continuing with step 10:
- If you are unsure of the location of the existing server directory, you can find it as follows:
ps -aef | grep radius
 - Change to the existing server directory.
 - Stop the server:
./S90radius stop
 - Unconfigure the previous installation:
sh install.sh -unconfig
 - The script prompts you to enter the path to the existing server directory:
Enter server directory [*current_directory/radius*]:
 - Type the path and press **[Enter]**.
 - Change to the directory into which you want to install the new software.
 - Run the `install.sh` script with the **-all** option:
sh install.sh -all

- 10 The `install.sh` script checks for Steel-Belted Radius configuration files:

```
Checking for previous configuration files
```

If the script detects existing configuration files, the following prompt is displayed; if not, you can skip to step 11:

```
Previous configuration files exist
Configuration files exist in server_directory
Do you want to discard them? [n]
```

If you type **n** (the default), a directory called **OLDCONFIG** is created under the server directory, and the previous configuration files is moved to this directory. If you type **y**, the installation script overwrites the previous files.

Warning: If existing configuration files are overwritten, any network-specific information defined there is lost. You should back up all files and subdirectories in the server directory before installing a new version of Steel-Belted Radius.

- 11 The `install.sh` script checks for Steel-Belted Radius database files:

```
Checking for database files in server_directory
```

If the script detects existing database files, the following prompt is displayed; if not, you can skip to step 12:

```
Previous database files exist
Database files exist in server_directory
Do you want to overwrite them? [n]
```

If you type **n** (the default), the database files are not be overwritten and the new version of Steel-Belted Radius has the entire administrative database (RAS Clients, Users, and so forth) from the previous installation.

If you type **y**, the database files is overwritten and all data previously contained in those files is lost.

Warning: For this reason, you should back up all files and subdirectories in the server directory before installing a new version of Steel-Belted Radius.

- 12 The `install.sh` script copies files to the server directory.
- 13 The `install.sh` script prompts you for the directory where you want to install the Steel-Belted Radius administration program and online help; this is known as the *admin directory*. Enter a full pathname.

If the admin directory does not already exist, the script creates it. The script then copies administration and help files to this directory.

- 14 Steel-Belted Radius includes a Java administration program that can be run locally from a browser on your UNIX machine, served up from a web server, or run remotely from a PC on your network. Consider how you would like to use this program, and set up the method of your choice.

See the “Setting Up the Administrator Program” section below.

Steel-Belted Radius also provides an LDAP interface that permits command line access to the Steel-Belted Radius administration database.

See “LDAP Configuration Interface” on page 332.

- 15 You can now configure Steel-Belted Radius by running the `install.sh` script with the **-config** option from the server directory, as follows:

sh install.sh -config

- 16 This command prompts you to configure Steel-Belted Radius for use with SNMP:

```
Do you want to configure SNMP? [n]:
```

If no, press **[Enter]** to proceed to the next prompt. If yes, type **y** and press **[Enter]**. you are prompted to specify file locations for SNMP:

```
Enter path for SNMP configuration files.
```

```
[/etc/snmp/conf]:
```

```
Enter path for SNMP library files. [/usr/lib/snmp]:
```

The default settings are usually correct; to accept them press **[Enter]** at each prompt.

- 17 The script prompts you to configure Steel-Belted Radius for use with an external SQL database.

Note: Steel-Belted Radius can be guaranteed to work only with Oracle versions 7 and 8.

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

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

If yes, type **y** and press **[Enter]**. you are prompted for each supported SQL vendor’s database in turn. If you type **y** to configure any one of them, you are then prompted to enter the required library paths. For example:

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

```
Supported Oracle versions: 7,8
```

```
What version of Oracle will be used? [8]:
```

Note: You need provide only the whole number for the version of Oracle you are running (i.e., a response of 7 works for 7.1, 7.2, etc.).

```
Setting the environment variable ORACLE_HOME
```

```
Enter ORACLE_HOME []:
```

```
Setting the environment variable LD_LIBRARY_PATH
```

```
Enter the path for Oracle shared libraries [xxx/lib]:
```

```
Setting the environment variable TNS_ADMIN
```

```
Enter TNS_ADMIN [xxx/network/admin]:
```

- 18 The script prompts you to configure Steel-Belted Radius for use with an external LDAP database.

Do you want to configure LDAP? [n]:

If no, simply press **[Enter]** to complete the **install.sh -config** script.

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]:

The default path /usr/lib is usually correct; to accept this default press **[Enter]** at the prompt.

Note: If you configure for LDAP emulation with the config.sh file, you must also explicitly set Enable=1 in the radius.ini file.

- 19 The script now runs as configured in the steps above. When it completes configuration, it displays the following message:

```
Admin configuration completed
```

- 20 To start the radius daemon without waiting for the next system restart, change to the server directory that you chose while installing Steel-Belted Radius, and start the server as follows:

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

Note: If your Solaris system is configured for shadow mode and you plan to use /etc/passwd authentication, you must run the radius daemon as root. You must also run the radius daemon as root if you use a port whose number is less than 1024.

- 21 You must now finish configuring the new Steel-Belted Radius server to suit your network's authentication and accounting needs.

See "Configuring the Server" on page 19.

Upgrading from a Previous Installation

We recommend that you do the following whenever you upgrade a UNIX installation:

- 1 Stop the previous version of the radius daemon.
- 2 Back up the Steel-Belted Radius server directory.
- 3 Complete the installation procedure as described above.

Restoring the Previous Configuration

Upon installation, old configuration files can be saved to a directory named `OLDCONFIG`. This practice prevents the loss of configuration information from a previous installation.

The files that are saved to the `OLDCONFIG` directory are:

- All the `.ini` files (`vendor.ini`, `account.ini`, `radius.ini`, and so forth)
- Any `.aut` files (external authentication method (e.g., SQL or LDAP) initialization files)
- Any `.acc` files (external accounting method (e.g., SQL) initialization files)
- Any `.pro` or `.dir` files (realm configuration files)

If you do not want to install any new features, copy all the files from the `OLDCONFIG` directory into the Steel-Belted Radius server directory. If you want to install the new features while maintaining current configuration information, complete the following tasks:

File	Task
<code>account.ini</code>	Modify <code>account.ini</code> in the server's directory with any changes you made in <code>OLDCONFIG/account.ini</code> .
<code>bounce.ini</code>	Copy <code>OLDCONFIG/bounce.ini</code> to the server directory
<code>filter.ini</code>	Copy <code>OLDCONFIG/filter.ini</code> to the server directory.
<code>proxy.ini</code>	Copy <code>OLDCONFIG/proxy.ini</code> to the server directory.
<code>radius.ini</code>	Modify <code>radius.ini</code> in the server directory with any changes you made in <code>OLDCONFIG/radius.ini</code> .
<code>tacplus.ini</code>	Copy <code>OLDCONFIG/tacplus.ini</code> to the server directory.
<code>vendor.ini</code>	Insert additional settings from <code>vendor.ini</code> in the server directory into <code>OLDCONFIG/vendor.ini</code> and copy this file to the server directory.
pro files	Insert additional settings from the sample.pro file in the server directory into each pro file saved in the <code>OLDCONFIG</code> directory and copy these files to the server directory.
acc files	Insert additional settings from the <code>radsql.acc</code> file in the server directory into the <code>radsql.acc</code> file (and each additional saved acc file) saved in the <code>OLDCONFIG</code> directory and copy these files to the server directory

File	Task
aut files	<p>Insert additional settings from the radsq1.aut file in the server directory into the radsq1.aut file (and each additional saved SQL aut file) saved in the OLDCONFIG directory and copy these files to the server directory.</p> <p>Insert additional settings from the ldapauth.aut file in the server directory into the ldapauth.aut file (and each additional saved LDAP aut file) saved in the OLDCONFIG directory and copy these files to the server directory.</p>
eap files	<p>Insert additional settings from each .eap file in the server directory into the corresponding file saved in the OLDCONFIG directory and copy these files to the server directory.</p>

Stopping and Starting the radius Daemon

After it is installed on the server, the radius daemon stops and starts automatically each time you shut down or boot up the server.

You can stop the radius daemon at any time. Change to the server directory that you chose at installation time, and stop the daemon, as follows:

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

To start the radius daemon:

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

Note: If your Solaris system is configured for shadow mode and you plan to use pass-through authentication to UNIX users and groups, you need to start the radius daemon under the root directory.

Setting Up the Administrator Program

The Steel-Belted Radius Administrator program runs as a Java applet within a browser. This program allows you to populate and configure your Steel-Belted Radius server via a graphical user interface. While running the Administrator, you can view online help, get RAS configuration tips, add a license key, import or export RADIUS data, and report on the server configuration.

You can deploy the Java Administrator in various ways. For example:

- You can run the Java administrator from within a browser installed on the local UNIX machine. Simply launch the browser, browse your local file

system and select either the default.htm or the index.html file. Both of these files reside in the java subdirectory under the admin directory (usually found under /radadmin/java).

- You can make the program accessible via a web server by doing one of the following:
 - Move (copy or FTP) the Java UI files from the java subdirectory to a part of the file system accessible to the web server.
 - Add a symbolic link pointing to the Java UI folder to a folder that is accessible to the web server. The symbolic link should point to the java subdirectory.
- You can simply transfer (FTP) the Java UI files from the java subdirectory to a PC on your network and run the program from within a browser installed on that PC. Be sure to move the entire java subdirectory to the PC, keeping the directory structure intact. After the move is complete, select either the default.htm or the index.html file. This launches the browser and the Java administrator applet.

Installing under Windows

This section describes how to install the Steel-Belted Radius service onto a Windows domain controller, server, or workstation.

Before You Begin

Your first task is to choose an appropriate Windows NT/2000 machine on which to install Steel-Belted Radius. The machine's role in your network affects the types of Windows NT security against which it can authenticate users.

If Steel-Belted Radius is installed on:

- A Windows NT/2000 server that is a Domain Controller, only Domain authentications are possible. You must give all of your remote access users the `Log On Locally` privilege on the Domain Controller machine.

Note: If you are not using Windows domain authentication (Winauth.dll), you must run Steel-Belted Radius on a Domain Controller if you want to use the MPPE keys to support a PPTP tunnel.

- A Windows NT/2000 server or workstation that is a member of a Domain, both Domain and Host authentications are possible.

- A Windows NT/2000 server or workstation that is not a member of a Domain, only Host authentications are possible.

Note: Authentication methods that do not involve pass-through to local platform security (Native, Proxy, SecurID, SQL, and so on) are available regardless of the server's role in the network.

Installing Steel-Belted Radius

- 1 Review the system requirements described in “System Requirements” on page 3.
- 2 If you have installed a previous version of this same edition of Steel-Belted Radius on this computer, please read the “Upgrading from a Previous Installation” on page 15.
- 3 Log into the Windows NT/2000 server or workstation. Insert the Steel-Belted Radius installation disk, choose **Start > Run**, and enter the drive letter and **Setup** command. For example:
D:\SETUP *
- 4 When the License Key screen appears, enter the License key printed on your license agreement card, or check the **Install 30-day trial** box.
Click **Next** to continue.
- 5 If you check the **Install 30-day trial** box, another License Key screen appears. Select the version of the product you want to evaluate.
Click **Next** to continue.
- 6 The Software License Agreement screen appears. Before proceeding, make sure that you read and agree with the terms of the license agreement.
Click **Yes** to indicate your agreement and to continue.
- 7 The Welcome screen appears.
Click **Next** to continue.
- 8 The Select Components screen appears. For a normal installation, make sure both **RADIUS Admin Program** and **RADIUS Server** are checked. You can use the default destination directory for each component you are installing, or you can click **Browse** to select a different directory.
Click **Next** to continue.
- 9 If you are upgrading from a previous installation, a warning screen appears, indicating that existing configuration files will be moved to the Service\Old directory.

Click **OK** to continue.

- 10 The Select Program Folder screen appears. You can accept the default folder (Steel-Belted Radius) or enter a different folder name.

Click **Next** to continue.

- 11 The server is installed under LocalSystem.

Click **Next** to continue.

- 12 The Start Copying Files screen displays the current settings for the installation. Scroll down and make sure the settings are exactly as you want them.

If the settings are correct, click **Next** to proceed with installation. Otherwise, click **Back** to return to previous screens.

- 13 After installation is completed, the Setup Complete screen appears. This screen gives you the opportunity to view the readme.txt file and start the Steel-Belted Radius Administrator.

Check the options you want to select, and click **Finish**.

- 14 Configure the new Steel-Belted Radius server to support your network's authentication and accounting needs.

You are now ready to configure the Steel-Belted Radius server. For more information, see "Configuring the Server" on page 19.

Upgrading from a Previous Installation

We recommend that you do the following when you upgrade a Windows NT/2000 installation:

- 1 Stop the previous version of the Steel-Belted Radius service. For more information, see "Starting and Stopping the RADIUS Service" on page 17.
- 2 Back up the Steel-Belted Radius server directory.
- 3 Complete the Steel-Belted Radius installation procedure as described above.

Restoring a Previous Configuration

When Steel-Belted Radius is installed, configuration files from a previous installation are archived to the Service\Old directory. This preserves configuration information for future reference.

The following files are archived to the Service\Old directory:

- *.ini files, such as vendor.ini, account.ini, and radius.ini

- *.aut files (external authentication method (e.g., SQL or LDAP) initialization files)
- *.acc files (external accounting method (e.g., SQL) initialization files)
- *.pro or *.dir files (realm configuration files)

If you do not want to install any new features, copy all the files from the Service\Old directory into the Steel-Belted Radius server directory. If you want to install the new features while maintaining current configuration information, complete the following tasks:

File	Task
account.ini	Modify account.ini in the server directory with any changes you made in Service\Old\account.ini.
bounce.ini	Copy Service\Old\bounce.ini to the server directory.
events.ini	Copy Service\Old\events.ini to the server directory.
filter.ini	Copy Service\Old\filter.ini to the server directory
proxy.ini	Copy Service\Old\proxy.ini to the server directory
radius.ini	Modify radius.ini in the server directory with any changes you made in Service\Old\radius.ini
tacplus.ini	Copy Service\Old\tacplus.ini to the server directory
vendor.ini	Insert additional settings from vendor.ini in the server directory into Service\Old\vendor.ini and copy this file to the server directory
pro files	Insert additional settings from the sample.pro file in the server directory into each pro file saved in the Service\Old directory and copy these files to the server directory
acc files	Insert additional settings from the sqlacct.acc file in the server directory into the sqlacct.acc file (and each additional saved acc file) saved in the Service\Old directory and copy these files to the server directory
aut files	<p>Insert additional settings from the sqlauth.aut file in the server directory into the sqlauth.aut file (and each additional saved SQL aut file) saved in the Service\Old directory and copy these files to the server directory.</p> <p>Insert additional settings from the ldapauth.aut file in the server directory into the ldapauth.aut file (and each additional saved LDAP aut file) saved in the Service\Old directory and copy these files to the server directory.</p>

Starting and Stopping the RADIUS Service

Steel-Belted Radius runs as an NT service. By default, it is set to run automatically whenever you start up Windows NT/2000.

If you don't want Steel-Belted Radius to run automatically, choose **Services** from the Control Panel, select **Steel-Belted Radius** from the **Service** list, click **Startup...**, and set the **Startup Type** to **Manual**. You can then use the **Start** and **Stop** buttons to control when Steel-Belted Radius runs.

Configuring for Authentication against Remote Domains

If you want to authenticate against Domains on Windows NT4, you must establish the proper one-way "trust."

At a minimum, there must be a one-way trust established between the Domain(s) on which Steel-Belted Radius is installed (the *Resource Domain*) and the *Remote Domain(s)* that contain(s) the usernames against which Steel-Belted Radius will authenticate. Each Remote Domain that contains user information must be "trusted" by the Resource Domain. A *trusting domain* is defined as the Resource Domain that trusts a Remote Domain; a *trusted domain* is a Remote Domain trusted by a Resource Domain.

Before establishing a Trust Relationship, the Administrator must identify the Resource Domain and the Remote Domains.

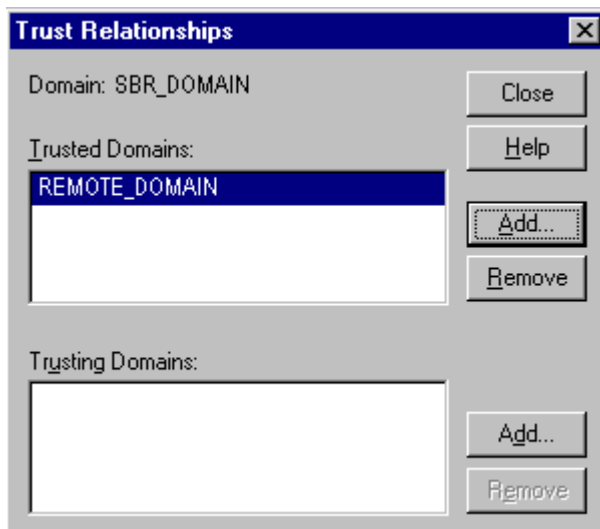
In the following example, the (Local) Resource Domain on which Steel-Belted Radius has been installed is called SBR_DOMAIN. The Domain that has Domain Users and/or Global Groups that Steel-Belted Radius must be configured to verify against is called REMOTE_DOMAIN.

Do This On the Remote Domain

- 1 Bring up the User Manager dialog on the Remote Domain (**Start > Programs > Administrative Tools > Active Directory Domains and Trusts**).
- 2 Right-click the local domain and choose **Properties** from the context menu.
- 3 Select the **Trusts** tab.
- 4 Select the option to **Add... SBR_DOMAIN** to the list of **Trusting Domains**.
- 5 Follow instructions to add SBR_DOMAIN as the Trusting Domain, including specifying a password to be used for this one-way Trust.

Now Do This On The Resource Domain

- 6 Bring up the User Manager dialog on the Resource Domain (**Start > Programs > Administrative Tools > User Manager**).
- 7 Go to the Trust Relationships dialog for SBR_DOMAIN.



- 8 Choose **Add...** to add REMOTE_DOMAIN to the list of **Trusted Domains**, supplying the password you defined in step 4.

Upgrading from a 30-Day Trial Installation

If you've downloaded Steel-Belted Radius on a 30-day trial basis and want to continue using the product, you do not need to re-install the software. All you need to do is add a license key to your existing installation.

First, purchase the Steel-Belted Radius software, either by contacting your preferred reseller or by contacting Funk Software directly. You will be shipped a product package that contains a license key.

Next, add the license key, as instructed below. The license key converts your 30-day trial software to an unlimited version.

Adding a License Key

Depending upon your purchasing arrangements, your Steel-Belted Radius software may require a new license key at some point after its initial installation.

If you are given a new license key by your reseller or by Steel-Belted Radius, you can add the key to an existing Steel-Belted Radius installation as follows:

- 1 Start the Steel-Belted Radius Administrator program and connect to the server.
- 2 For **UNIX**, click the **License** button at the lower right of the main window.
For **Windows**, select **File > License**.
- 3 The Add a License for Server dialog appears. Enter the license key and click **OK**. If you are running Windows and the license key you've entered is invalid, the server displays an error message; click **OK** in this message box and try again.
- 4 After you've entered a valid license key, the server displays a confirmation message and reminds you that you must restart the server. When you click **OK** in this message box, the server does not restart itself automatically; you'll need to restart it manually.
- 5 The next time Steel-Belted Radius is started, the new license is loaded.

Configuring the Server

After you've installed the Steel-Belted Radius software on your computer, and have added the appropriate license keys, you must configure the software before it can be used.

The specific steps you must perform depend on your network's authentication and accounting needs. However, the basic steps are:

- 1 Make sure the computer on which you're running Steel-Belted Radius has the IP protocol configured.
- 2 Configure each of your NAS devices to communicate with the server. To do this, you must log into each device and run its configuration interface.
- 3 Run the Steel-Belted Radius Administrator program.
- 4 Using the Servers dialog, connect to your server (under **UNIX** you do this by using the default account **admin** and password **radius**).

- 5 **UNIX only:** Using the Access dialog, change the default administrative account password from **radius** to a password of your choosing.
- 6 Using the RAS Clients dialog, configure the server to communicate with each of its RADIUS clients (NAS devices).
- 7 From the Users dialog, identify each of the users or groups of users that are permitted to dial in to the NAS devices. Select user attributes, either by assigning them in the Users dialog or by creating user profiles in the Profiles dialog.

These are the basic steps required by most configurations. Next, you should consider the unique features required by your configuration. For example:

- If you plan to use the auto-restart feature (the server software restarts itself automatically whenever it experiences a shutdown) you must edit the related configuration file(s).
For **UNIX**, see “Auto-Restart Files (UNIX only)” on page 249.
For **Windows**, see “bounce.ini File (Windows only)” on page 192.
- If you plan to configure administrative access levels, you must edit the related configuration files.
See “access.ini File” on page 177.
See also “admin.ini File” on page 184.
- If you plan to use Proxy RADIUS, you must identify the Proxy RADIUS realms to which the server forwards authentication or accounting messages, and identify the target servers within each realm.
See “Proxy RADIUS” on page 65.
See also “Configuring a Proxy RADIUS Realm” on page 257.
- If you plan to use the LDAP configuration interface with Steel-Belted Radius, you must install LDAP command line utilities and configure LDIF files.
See “LDAP Configuration Interface” on page 332.
- If you run UNIX and plan to use external databases for authentication or accounting purposes and did not configure this feature when prompted by the Steel-Belted Radius installation script, you must do so before this feature can work.
See “Configuring External Databases” on page 21.
- If you plan to use SecurID authentication, you must configure Steel-Belted Radius to communicate with the ACE/Server.
See “Configuring SecurID Authentication” on page 24.

- If you plan to use TACACS+ authentication, you must configure Steel-Belted Radius to communicate with the TACACS+ server.
See “Configuring TACACS+ Authentication” on page 26.
- If you plan to use directed authentication or accounting methods, you must configure the realm with which each method is associated.
See “Directed Authentication” on page 37 and “Directed Accounting” on page 51. See also “Configuring a Directed Realm” on page 263.

UNIX-Specific Configuration

Configuring External Databases

If you plan to use external databases for authentication or accounting, you do not need to configure this feature when prompted by the Steel-Belted Radius installation script. You can configure this feature after Steel-Belted Radius has been installed on the UNIX server or workstation.

The steps are as follows:

- 1 Optionally, perform the instructions in “SQL Authentication” on page 364 and/or “SQL Accounting” on page 388.
- 2 If you want to use Steel-Belted Radius with an LDAP database, consult your LDAP database vendor’s documentation.
- 3 Perform the instructions in “External LDAP Authentication” on page 404.
- 4 Note that if you run the `install.sh` script at any time after installing the current version of Steel-Belted Radius, you must run it as follows:
 - Stop the Steel-Belted Radius server. Change to the directory where the `install.sh` script resides and enter:
sh install.sh -unconfig
 - This prompts you to enter the path to the existing server directory.
Enter server directory [*current_directory/radius*]:
Type the path and press **[Enter]**.
 - Now that you have stopped and unconfigured the existing server, run the `install.sh` script with the **-config** option from your current directory:
sh install.sh -config

Configuring SNMP Support

To configure Steel-Belted Radius for SNMP support:

The Solstice Enterprise Agent (SEA) from Sun Microsystems is required for support of SNMP in Steel-Belted Radius. You can download the SEA software from the Sun Microsystems web site (<http://www.sun.com>).

- 1 Install the SEA software.

You must install the `SUNWsacom` and `SUNWsasnm` packages. Optionally, you can install the `SUNWmibii` package, which installs Sun's MIB II Sub Agent. These packages are included in the Runtime version of SEA.

Be sure to download the appropriate compressed file for your version of Solaris. Note that the SEA installation instructions may require you to remove existing SEA packages in sequence before adding the new SEA packages.

- 2 Reboot your UNIX system to load the Master Agent (`snmpdx`).
- 3 Run the Steel-Belted Radius `install.sh -config` script.

Note: If you've already installed the current version of Steel-Belted Radius, it is not necessary to re-install the software before you configure SNMP. However, if SEA was not installed at the time that you installed Steel-Belted Radius, you must run `install.sh -unconfig` first, then `install.sh -config`.

Do the following to ensure that the SNMP Sub Agent is configured:

- When prompted to configure SNMP, type **Y** and press **[Enter]**.
When prompted for the location of the SNMP configuration files, specify the location to which you installed the package or accept the default (`/etc/snmp/conf`). The `radsnmp.reg`, `radsnmp.rsrc`, and `radsnmp.acl` files are copied from the Steel-Belted Radius SNMP configuration file directory to this location. The `radsnmp.rsrc` file is modified to include the path to the Steel-Belted Radius server directory. The `radsnmp.reg` file registers the Steel-Belted Radius Sub Agent with the Master Agent the next time the Master Agent starts up.
- When prompted for the path to the SNMP library files, specify the location to which you installed the package or accept the default filepath (`/usr/lib/snmp`). The `radsnmp` Sub Agent is copied from the Steel-Belted Radius SNMP configuration file directory to this location.

Note: The default file locations should usually be accepted.

The installation script configures SNMP support by copying the relevant files. It also creates a file to contain TCP configuration data (`radsnmp.inf`). Both files are placed in the same directory as the radius daemon.

Note: When you start the radius daemon at the end of the Steel-Belted Radius installation procedure, the `yyyymmdd.LOG` file and UNIX command shell window both indicate that "SNMP is Enabled." This message appears whenever a license key for Steel-Belted Radius is present. It does not confirm that SNMP is correctly configured.

4 If you want to enable support for SNMP traps, you must perform the following steps. If you do not want to enable support for SNMP traps, skip these steps and move on to step 4.

- Change to the `/etc/snmp/conf` directory.
- Edit `enterprises.oid` to include the following entry:

```
"funktSbrTraps""1.3.6.1.4.1.1411.1.1"
```

This syntax must be exact. Save the change.

- Edit `snmpdx.acl` to include the following entries in the Trap Parameters `trap = {` section:

```
trap-community = SBR-trap
hosts = host_name_of_SNMP_manager_machine
enterprise = "funktSbrTraps"
trap-num = 100-115, 5000-5029, 10000-10052
```

Note: Be sure to place the '{' and '}' brackets appropriately for each section in `snmpdx.acl`. Use the existing sample entries as a model if necessary. Do not edit the trap-recipients section at the bottom of `snmpdx.acl`.

5 Stop and restart the SNMP Master Agent (`snmpdx`). This can be done by changing to the `/etc/rc2.d` directory and running the following commands:

```
./K07snmpdx stop
./K07snmpdx start
```

Stopping and restarting the Master Agent causes it to load the `radsnmp` Sub Agent.

6 Verify that the Master Agent and Sub Agent are both running on the Steel-Belted Radius server. You can verify that the Sub Agent is running as follows:

```
ps -aef | grep radsnmp
```

You can verify that the Master Agent is running as follows:

```
ps -aef | grep snmpdx
```

7 If you do not already have an SNMP Manager on your network, install one. This can be any SNMPv1-compliant Manager software.

- 8 Load the Steel-Belted Radius MIBs into your SNMP Manager software. This lets you select the SNMP variables (the specific supported queries) that can be sent to the Steel-Belted Radius server. Each SNMP Manager loads the MIBs differently.

See your SNMP Manager documentation for details.

The MIBs are found in the `snmp` subdirectory under the Steel-Belted Radius server directory (usually `/radius/snmp`). The file names are:

- `rauths.mib` for RADIUS authentication
- `raccs.mib` for RADIUS accounting
- `fnkradtr.mib` for SNMP traps and alarms

Configuring SecurID Authentication

Perform the following steps to configure a Steel-Belted Radius server to work with an ACE/Server. If you are not familiar with the ACE/Server from Security Dynamics, Inc. then you might want to contact your ACE administrator for assistance. Otherwise, you can follow these steps yourself:

- 1 Copy the `sdconf.rec` file from its usual location on the ACE/Server (`\ACE\data`) to the appropriate directory:
 - On **Windows**: `C:\winnt\system32`
 - On **UNIX**: the directory that contains the radius daemon on the Steel-Belted Radius server.

If you copy the file after the Steel-Belted Radius service, or daemon, has been started, you must stop and start Steel-Belted Radius before SecurID can work.

- 2 Verify that the Steel-Belted Radius server has an entry on the ACE/Server.
 - a Start the ACE/Server administration program and display the list of clients.
 - b If the list of clients does not include the Steel-Belted Radius server, select **Client > Add Client** and complete the Client dialog, giving the Steel-Belted Radius server a Client type of Net OS Client.
- 3 Verify connectivity as follows: The ACE/Server offers a monitoring window on which it logs every authentication transaction, complete with the reason for the accept or reject decision. You can verify that pass-through to SecurID is working, by creating a SecurID User called `<ANY>` and then attempting to access the network. Look for your request on the ACE/Server monitor screen. If access is denied, you'll know that there's a configuration problem. Try these steps again, or contact your ACE administrator for assistance.

- 4 Edit the [SecurID] section of radius.ini. This initialization file is found in the same directory as the Steel-Belted Radius service (for **Windows**, usually C:\RADIUS\Service) or daemon (for **UNIX**).

Ensure that the CachePasscodes field is set to yes and the SecondsToCachePasscodes field is set to an appropriate number of seconds. These settings ensure that authenticated SecurID users can open a second B-channel during an ISDN connection.

See “radius.ini [SecurID] Section” on page 221.

- 5 Edit the [SecurID] section of the eap.ini file. This initialization file is found in the same directory as the Steel-Belted Radius service (for **Windows**, usually C:\RADIUS\Service) or daemon (for **UNIX**).

Ensure the EAP settings in this section are enabled (remove semi-colon character from the front of each line) if you plan to use SecurID authentication with EAP Generic-Token protocol support. The client system must support this protocol as well for this combination to work.

- 6 If you edit the radius.ini or eap.ini files after Steel-Belted Radius has been started, then you must stop and restart Steel-Belted Radius before your changes take effect.

These steps complete initial setup of the two servers. To fully enable pass-through authentication to the ACE/Server, you must also set up the SecurID authentication method.

See “Configuring Authentication Methods” on page 38.

See also “Adding a SecurID User” on page 107.

Configuring Location of the sdconf.rec File (UNIX only)

The variable VAR_ACE in the S90radius script file lets you specify the directory holding the sdconf.rec file. (This variable must also be “exported” so that Steel-Belted Radius can use it.)

For example:

```
VAR_ACE="$RADIUSDIR/ace"  
export VAR_ACE
```

This variable is set by default in the file to point to the radius directory. If the variable is not set at all in the file, the server sets the value of this variable to /var/ace.

Configuring TACACS+ Authentication

To configure a Steel-Belted Radius server to work with a TACACS+ server, the following steps must be performed:

- 1 The `tacplus.ini` file must be present in the same directory as the Steel-Belted Radius service (in the case of Windows, usually `C:\RADIUS\Service`), or daemon (in the case of UNIX). This happens automatically following installation.
- 2 You must edit `tacplus.ini` to identify the shared secret and host machine that you use for TACACS+. See “`tacplus.ini [ServerInfo] Section`” on page 231.
- 3 If you edit `tacplus.ini` after Steel-Belted Radius has been started, then you must stop and restart it before your changes take effect.

These steps complete initial setup of the two servers. To fully enable pass-through authentication to the TACACS+ server, you must also set up the TACACS+ authentication method.

See “Configuring Authentication Methods” on page 38.

See also “Adding a TACACS+ User” on page 109.

Oracle Software Changes

Any time that your Oracle software changes by upgrading, regressing (reverting to an earlier version), or removal, you must ensure that the corresponding links in Steel-Belted Radius are updated. It is likely to crash or refuse to run if it remains configured for a version of Oracle it cannot find.

Important: Use only one version of Oracle at a time. Never mix different versions of Oracle.

If your Oracle software changes, you should follow one of the following procedures:

- Issue the following commands to Steel-Belted Radius via the command line:

```
sh config.sh -unconfig
sh config.sh -config
```

Then answer the questions appropriately.

- Alternatively, if your Oracle software has changed because of an upgrade, you can stop the Steel-Belted Radius server and manually recreate the following symbolic links in the server directory:

Link	Becomes
radius	radius_generic if Oracle is not used radius_ora7 if Oracle version 7 is in use. radius_ora8 if Oracle version 8 is in use.
radsql_acct_ora.so	radsql_acct_ora7 if Oracle version 7 is in use. radsql_acct_ora8 if Oracle version 8 is in use.
radsql_auth_ora.so	radsql_auth_ora7 if Oracle version 7 is in use. radsql_auth_ora8 if Oracle version 8 is in use.

Concepts

3

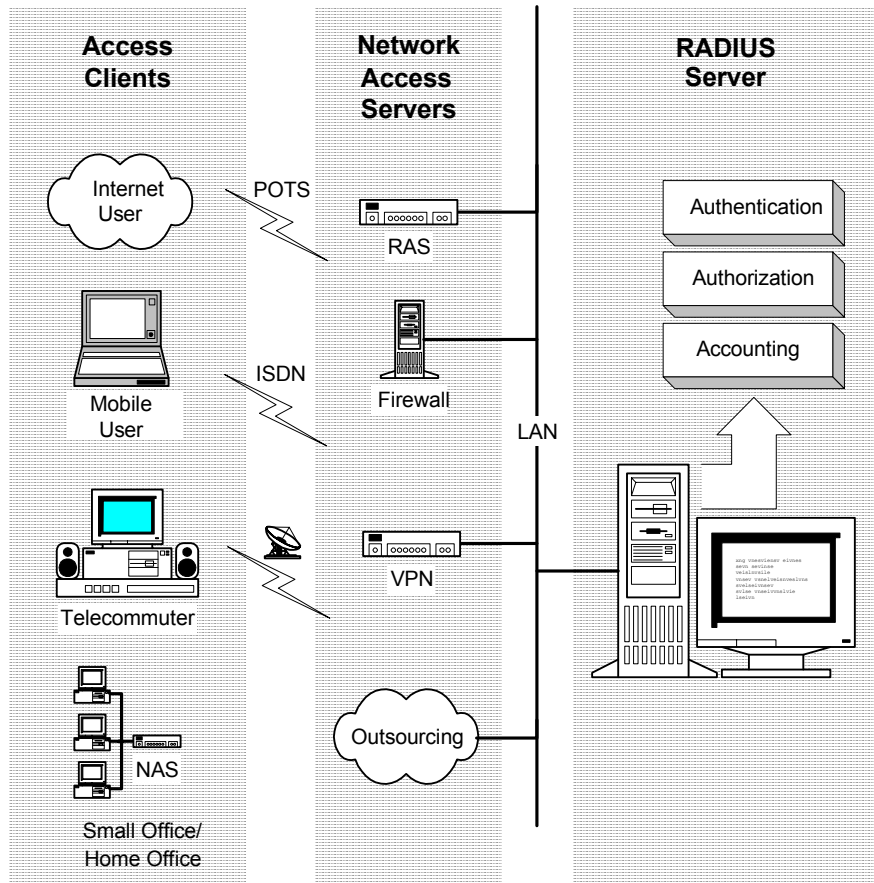
- RADIUS Basics
- Authentication
- Password Protocols
- Accounting
- Attributes
- Profiles
- Request Routing
- Proxy RADIUS
- Tunnels
- IP Address Assignment
- Resource Management
- Technical Bulletins

RADIUS Basics

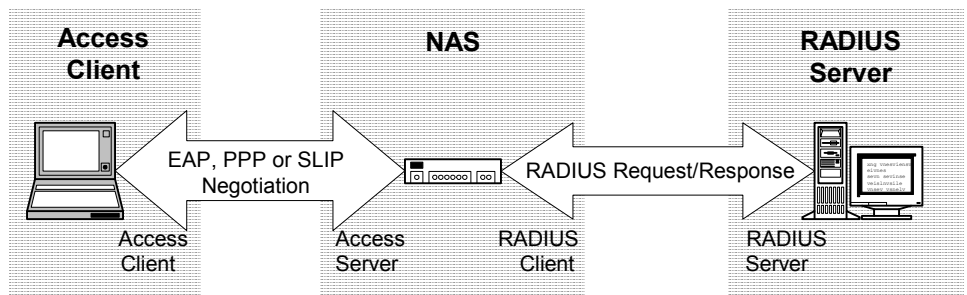
RADIUS (Remote Authentication Dial In User Service) is an industry-standard protocol for providing authentication, authorization, and accounting services. A RADIUS *access client* (such as a remote office, remote user, or mobile user with dialup or wireless network access) sends an authentication request containing identification and connection information to a *network access server* (NAS).

The NAS is a device that can recognize and handle connection requests from outside the network “edge,” such as a wireless Access Point, an ISDN bridge, or a modem pool. When the NAS receives a user’s connection request, it might perform an initial access negotiation with the user (EAP, PPP or SLIP) to establish certain data (username, password, NAS device identifier, NAS port number, and so on). The NAS then passes this data to the *RADIUS server* and requests authentication.

The RADIUS server authenticates the request and authorizes services over the connection by matching data from the NAS’s request with entries in some well-known, trusted database. In the case of Steel-Belted Radius, the match might be found on the RADIUS server; on some other type of authentication server (ACE/Server or TACACS+); in a SQL or LDAP database; or on some other RADIUS server for which this server is a *proxy*.



RADIUS-Based Remote Access Environment



Data Exchange between Access Client, NAS, and RADIUS server

If a match is found, the RADIUS server accepts the user. If a match is not found, it rejects the user. Based on the response from the RADIUS server, the NAS decides

whether to establish the user's connection or terminate the user's connection attempt. Finally, the NAS forwards accounting data to the RADIUS server to document the transaction; the RADIUS server can store or forward this data to support billing for the services provided.

RADIUS Packets

A RADIUS client and RADIUS server communicate by means of RADIUS packets. RADIUS packets are formatted using conventions outlined in RFC 2865, "Remote Authentication Dial In User Service (RADIUS)" and RFC 2866, "RADIUS Accounting."

To configure the Steel-Belted Radius server, you need to know the following about RADIUS packets:

- They carry messages between the RADIUS client and RADIUS server.
- They follow a request/response convention: the client sends a request and expects a response from the server. If the response doesn't arrive, the client can retry the request periodically.
- Each packet supports a specific purpose: authentication or accounting.
- A packet can contain values, called *attributes*.
- The specific attributes to be found in each packet depend upon the type of packet (authentication or accounting) and the device that sent it (for example, the specific make and model of NAS device).

For more information on RADIUS authentication packet structures and attributes, see RFC 2865. For more information on RADIUS accounting packet structures and attributes, refer to RFC 2866.

RADIUS Configuration

You must configure a RADIUS client and RADIUS server before they can communicate. As shown in the diagram above, if the client is a NAS device, it's probably on the same LAN as the server. If so, the same network administrator probably has all the data and privileges necessary to configure both sides of RADIUS communications. Under other conditions, you might need to work out configuration details with the administrators of other networks.

RADIUS Server Configuration

You must tell a RADIUS server how to respond to each of its clients. When configuring the Steel-Belted Radius server, you'll need to start the Administrator

program, open the RAS Clients dialog, and enter the following information for each RADIUS client:

- The IP address of the client device;
- The RADIUS shared secret to be used by Steel-Belted Radius and the client device; *and*
- The make and model of the client device, selected from a list of devices that Steel-Belted Radius supports. If a specific make/model is not listed, select **- Standard Radius -**.

RADIUS also requires you to specify the UDP ports that you'd like the server to use when sending and receiving RADIUS authentication and accounting packets. Optionally, you can override the default port settings for Steel-Belted Radius.

See "RADIUS Ports" on page 35.

RADIUS Client Configuration

You must tell each RADIUS client how to contact its RADIUS server. When configuring a client to work with a Steel-Belted Radius server, you'll need to log into the client device, run its administration program, bring up its RADIUS configuration interface, and enter the following information:

- The IP address of the Steel-Belted Radius server
- The RADIUS shared secret to be used by the Steel-Belted Radius server and the client device
- The UDP ports on which the client device wants to send and receive RADIUS authentication and accounting packets. These must match the ports that Steel-Belted Radius is using for the same purposes.

Multiple RADIUS Servers

The RADIUS workload can be distributed among several servers, as follows:

- You can create specialized servers for RADIUS authentication and accounting services. To accomplish this, each client device must be configured to send its authentication packets to one RADIUS server and its accounting packets to another.
- You can provide redundancy by pairing RADIUS servers to work in tandem. Most NAS configuration interfaces permit you to designate primary and secondary servers for authentication and accounting.

If both measures for distributing the RADIUS workload are in place, client configuration involves naming the following for each client device: a primary

RADIUS accounting server, a secondary RADIUS accounting server, a primary RADIUS authentication server, and a secondary RADIUS authentication server.

RADIUS Shared Secret

The RADIUS *shared secret* is a case-sensitive password used to validate communications between two RADIUS devices. The shared secret can be any alphanumeric string. A shared secret must be configured identically on both sides of the RADIUS communication links. If your configuration includes a RADIUS client, a RADIUS proxy, and a RADIUS server, the shared secret used to validate communications between the client and proxy can be different from the shared secret used to validate communications between the proxy and server.

Configuring Shared Secrets

On the client side, most configuration interfaces allow you to enter different shared secrets for RADIUS authentication and RADIUS accounting purposes. If the interface also permits you to identify primary and secondary RADIUS servers, you can set up as many as four secrets (primary accounting, secondary accounting, primary authentication, and secondary authentication).

On the server side, the configuration interface allows you to create a list of known RADIUS clients (NAS devices). You should be able to identify the authentication shared secret and accounting shared secret that this server uses to communicate with each of the clients on this list.

See “RAS Clients Dialog” on page 87.

See also “Proxy Dialog” on page 112.

You should take steps to ensure that every shared secret is unique across your entire RADIUS configuration.

How Shared Secrets Are Used

During an authentication transaction, password information must be transmitted securely between the RADIUS client and the RADIUS server. Password security can be addressed using a variety of protocols such as PAP, CHAP, or MS-CHAP. When PAP is used, the password is encrypted and decrypted using the authentication shared secret.

See “Password Protocols” on page 44.

No encryption is involved in transmitting accounting data between a RADIUS client and RADIUS server. However, the accounting shared secret is used by each device

to verify that it can “trust” any RADIUS communications it receives from the other device.

RADIUS Ports

The RADIUS standard initially used UDP ports 1645 and 1646 for RADIUS authentication and accounting packets. The RADIUS standards group later changed the port assignments to 1812 and 1813, but many organizations still use the old 1645/1646 port numbers for RADIUS.

Any two devices that exchange RADIUS packets must use compatible UDP port numbers. That is, if you are configuring a NAS to exchange authentication packets with a RADIUS server, you must find out which port the server uses to receive authentication packets from its clients (1812, for example). You must then configure the NAS to send authentication packets on the same port (1812). The same is true for RADIUS accounting.

Steel-Belted Radius can listen on multiple ports. To provide maximum compatibility, the server listens to both old and new port standards by default. This means that, as a default setting, ports 1645 and 1812 are assigned to authentication and ports 1646 and 1813 are assigned to accounting. If you want to reassign ports, you can specify port numbers with the `UDPAuthPort` and `UDPAcctPort` settings in the [Ports] section of the `radius.ini` file, or edit the `services` configuration file.

See “services File” on page 246 and “radius.ini [Ports] Section” on page 220.

Authentication

To understand the authentication sequence, you’ll need an overview of RADIUS authentication messages. The following table explains the conditions under which each type of message is issued, and the purpose of any RADIUS attributes the message contains.

Message Conditions	Purpose of Message Attributes
When a NAS receives a connection request from a user, the NAS authenticates the request by sending an Access-Request to its RADIUS server.	Identify the user. Describe the type of connection the user is trying to establish.
When a RADIUS server is able to authenticate a connection request, it returns a RADIUS Access-Accept to its client (usually the NAS).	Allow the NAS to complete access negotiations. Configure connection details, for example, providing the NAS with an IP address that it can assign to the user. Enforce time limits and other “class of service” restrictions upon the connection.
When a RADIUS server is unable to authenticate a connection request, it returns an Access-Reject to its client (the NAS).	Terminate access negotiations. Identify the reason for failure.
If initial authentication conditions are met, but additional input is needed from the user, the RADIUS server returns an Access-Challenge to its client (the NAS).	Enable the NAS to prompt the user for more authentication data. Complete the current Access-Request, so the NAS can issue a new one.

Authentication Methods

Each time an Access-Request message arrives at the server, an authentication transaction begins. During this transaction, the server attempts to authenticate the request by trying each of its configured and enabled authentication methods in turn. To know which methods to try, and in which order, the server consults its Authentication Methods list. You can view and edit this list by starting the Administrator program and opening its Configuration dialog.

Native User Authentication

While trying the Native User method (for user accounts stored directly on the server itself), Steel-Belted Radius searches its database for an entry whose User-type is `Native User`, and whose User name matches the username in the Access-Request. If the entry:

- Cannot be found, or if it is found and the Password is invalid, Steel-Belted Radius tries the next enabled method in the Authentication Methods list.
- Is found, but its Check-List doesn’t match attributes found in the Access-Request, Steel-Belted Radius returns an Access-Reject to the NAS.

- Is found, and its Password and Check-List match perfectly, Steel-Belted Radius constructs an Access-Accept using the entry's Return-List, and returns it to the NAS.

Pass-Through Authentication

Pass-through authentication methods permit Steel-Belted Radius to begin the authentication by asking another entity to validate the username and password found in the Access-Request.

Steel-Belted Radius can pass-through to a Window NT security database, ACE/Server (SecurID), or TACACS+ server.

Proxy RADIUS Authentication

Steel-Belted Radius can convey an Access-Request to some other RADIUS server, which then (1) performs authentication according to its own conventions and (2) returns a response. Steel-Belted Radius then relays this response to the NAS. The set of conventions for relaying packets between cooperating RADIUS servers is known as *Proxy RADIUS*.

Note: Steel-Belted Radius offers a wide range of powerful authentication options which build upon Proxy RADIUS authentication and realms. See "Configuring a Proxy RADIUS Realm" on page 257.

External Authentication

External authentication methods permit Steel-Belted Radius to authenticate users by using configuration files that tell it how to (1) communicate with an external database, (2) query the database for authentication data, and (3) formulate its results into a response packet. Steel-Belted Radius then relays this response to the NAS. External authentication methods include SQL and LDAP.

See "SQL Authentication" on page 364 or "External LDAP Authentication" on page 404.

Directed Authentication

Every authentication request works its way through the same Authentication Methods list until one of the methods succeeds or the end of the list is reached.

This behavior might not be ideal for every account. If you want requests from certain users/accounts to bypass the master Authentication Methods list and use an alternate list, you can do so by employing the directed authentication feature. This feature allows you to map the User-Name or DNIS information in an incoming

authentication request to a specific list of authentication methods. The list can include any native, pass-through, proxy-as-authentication, or external database authentication method configured on the Steel-Belted Radius server.

You can also direct authentication towards a particular realm using a technique called *attribute mapping*. This allows you to check for the presence or absence of a particular attribute in an authentication request, or for it containing a specific value. Attribute mapping can be used with both Proxy Realms and Directed Realms.

See “Configuring a Directed Realm” on page 263.

Authenticate-Only Requests

Steel-Belted Radius also supports a request to simply authenticate a user. The NAS specifies this type of request by setting the `Service-Type` field to `Authenticate Only` (numeric value 8). The server responds with either an `Access-Reject` or an `Access-Accept` (without any attributes).

You can disable this feature (so that attributes are always returned in the response packet) by setting the `AuthenticateOnly` field in the [Configuration] section of the `radius.ini` file to 0.

See “radius.ini [Configuration] Section” on page 212.

Configuring the Authentication Sequence

After you configure authentication methods for the Steel-Belted Radius server, the Configuration dialog in the Administration application lists them in the **Authentication Methods** list. Methods appear in the order in which the server tries them. Names of enabled methods are displayed in black text; names of disabled methods are displayed in gray text. During an authentication transaction, the server works down the list, skipping disabled methods.

You can enable or disable methods, or re-order methods in the list, by using the control buttons in the Authentication Methods panel. In this way, you directly control the sequence of each authentication transaction.

See “Configuration Dialog” on page 134.

Configuring Authentication Methods

As we’ve seen, Steel-Belted Radius offers several authentication methods. Each method tries to find database entries that match the data in the incoming `Access-Request` packet. However, methods differ according to:

- The location of the database; *and*

- The conventions required to interact with the database.

Steel-Belted Radius configuration depends on the authentication methods you plan to use. The various tasks are summarized as follows.

Method	How to Configure	Complete Details
Native User	Create Native User entries in the Steel-Belted Radius database.	“Adding a Native User” on page 100
OS Pass-Through Security	This method assumes that you already have users, groups, and passwords defined in your local security database. Create User entries in the Steel-Belted Radius database. Choose User-types as appropriate.	“Users Dialog” on page 91
ACE/Server (SecurID)	This method assumes that you already have PIN/token code pairs defined on an ACE/Server. First, configure the Steel-Belted Radius server to communicate with the ACE/Server. Then create User entries in the Steel-Belted Radius database. Choose SecurID User, <ANY>, SecurID Prefix, and SecurID Suffix User-types.	“Configuring SecurID Authentication” on page 24 <i>and</i> “Adding a SecurID User” on page 107
TACACS+	This method assumes that you already have username/password pairs defined on a TACACS+ server. First, configure the Steel-Belted Radius server to communicate with the TACACS+ server. Then, create User entries in the Steel-Belted Radius database. Assign TACACS+ User, <ANY>, TACACS+ Prefix, and TACACS+ Suffix user-types.	“Configuring TACACS+ Authentication” on page 26 <i>and</i> “Adding a TACACS+ User” on page 109
Proxy RADIUS	<u>Add a single target</u> . You can set up single targets that are not associated with any realm. <i>or.</i> <u>Identify Proxy RADIUS realms</u> , each of which is a pool of Proxy RADIUS target servers. Each time a RADIUS request arrives addressed to this realm, Steel-Belted Radius dynamically selects the appropriate target within the realm.	“Proxy Dialog” on page 112 “Configuring a Proxy RADIUS Realm” on page 257

Method	How to Configure	Complete Details
External SQL Database	This method assumes that you already have User records stored in a SQL database. Create a Steel-Belted Radius .aut file that connects to a SQL database and issues a SELECT query based upon the username and password. Give the .aut file an InitializationString value of SQLName . Stop and restart the Steel-Belted Radius server. Subsequently, the SQL authentication method appears in the Configuration dialog's Authentication Methods list as <i>SQLName</i> . You can use the Configuration dialog to enable, disable, and re-order the <i>SQLName</i> method.	"Configuring SQL Authentication" on page 365
External LDAP Database	This method assumes that you already have User records stored in an LDAP database. Create a Steel-Belted Radius .aut file that validates the username and password based upon Bind and Search requests to an LDAP database. Give the .aut file an InitializationString value of <i>LDAPName</i> . Stop and restart the Steel-Belted Radius server. Subsequently, the LDAP authentication method appears in the Configuration dialog's Authentication Methods list as <i>LDAPName</i> . You can use the Configuration dialog to enable, disable, and re-order the <i>LDAPName</i> method as desired	"Configuring LDAP Authentication" on page 407

Method	How to Configure	Complete Details
EAP-TTLS	This method provides a means for an authentication request to be sent directly from the client to the server via a TLS connection. The act of establishing the TLS connection authenticates the server to the client and the authentication request sent through the tunnel authenticates the client to the server. Create a Steel-Belted Radius <code>ttlsauth.aut</code> file that specifies options for the TLS connection and the manner in which Steel-Belted Radius routes the inner authentication request. Stop and restart the Steel-Belted Radius server. Subsequently, the EAP-TTLS authentication method appears in the Configuration dialog's Authentication Methods list. You can use the Configuration dialog to enable, disable, and re-order EAP-TTLS methods.	"Configuring For EAP-TTLS and EAP-PEAP" on page 308
Directed Authentication	For each directed authentication method that you want to configure, add an entry to the [Directed] section of <code>proxy.ini</code> and create a <i>RealmName.dir</i> file that specifies the mapping between the routing information that you expect in the authentication packet, and a list of locally-configured authentication methods that you want to use.	"Configuring a Directed Realm" on page 263

Advanced Options

Steel-Belted Radius provides the following additional authentication control options:

Account Lockout

Account lockout allows you to disable an account after a configurable number of failed login attempts within a configurable period. For example, if a user enters an incorrect password three times within two minutes, Steel-Belted Radius can lock out the user's account temporarily. During the lockout period, the user cannot log in, even with the correct password.

When a user account is locked out, the user must wait until the expiration of the lockout period, or a network administrator can clear the lockout status for the account.

For information on configuring account lockout, see “lockout.ini File” on page 207. For information on clearing a locked-out account, see “Clearing Locked-Out Accounts” on page 207.

Important: *Do not enable account lockout and account redirection at the same time. If account lockout and account redirection are both enabled, account lockout is used and account redirection settings are ignored.*

Note: Account lockout state is not maintained if Steel-Belted Radius is restarted.

Account Redirection

Account redirection allows you to flag an account for special processing after a configurable number of failed login attempts within a configurable period. For example, if a user enters an incorrect password three times within two minutes, Steel-Belted Radius can accept the user (even with an incorrect password) but limit the user’s access to specific network resources, such as a secure webpage that prompts the user to provide other authentication information. If the user can obtain his or her current password (or can create a new one through such a secure web page), he or she can then reconnect and log in successfully.

When account redirection is enabled and a user repeatedly enters an incorrect password, Steel-Belted Radius places the user in *redirect* state. When a user is in redirect state:

- If the user does not submit another authentication request within a specified time-out period, the user is released from redirect state and returned to normal state.
- If the user submits another authentication request within a specified time-out period, the user is accepted without authentication/authorization processing. The accept message for the user includes the attributes and values specified in a redirection profile, and the user is placed into *accept-pending* state. An external customer process uses the attributes and values in the accept message to direct the user to an external process, which may prompt the user to enter alternate authentication information in order to receive a password via email.

When a user is in accept-pending state, the next authentication request received determines whether the user is accepted or locked out:

- If the user enters the appropriate authentication information, the user is returned to normal state and Steel-Belted Radius generates an informational SNMP trap message.
- If the user does not enter the appropriate authentication information, Steel-Belted Radius issues an Accept-Reject message, locks the user out of the network for a configurable lockout period, and generates an informational SNMP trap message. During this lockout period,

authentication requests for the user are automatically rejected, even if the user enters the correct password.

Optionally, you can identify RADIUS clients that you want to exclude from account redirection processing. Authentication requests from excluded RADIUS clients are processed normally, without use of redirection or account state changes.

For information on configuring account redirection, see “redirect.ini File” on page 228.

Important: *Do not enable account lockout and account redirection at the same time. If account lockout and account redirection are both enabled, account lockout is used and account redirection settings are ignored.*

Note: *Account redirection state is not maintained if Steel-Belted Radius is restarted.*

Blacklisting

This feature allows you to blacklist or automatically reject authentication requests that contain certain values. For example, the Calling-Station-Id attribute could be used to block users who dial in from a particular phone number.

Blacklisting functions on all local authentication requests and can be configured to include proxy-RADIUS requests.

See “blacklist.ini File” on page 191.

Allowed Access Hours

Steel-Belted Radius provides a vendor-specific attribute called `Funk-Allowed-Access-Hours`. This attribute can be placed in the Check-List for a User or profile entry to control the exact time periods during which a user can be allowed access. It can also be retrieved from a database backend.

During authentication, the server compares the value of `Funk-Allowed-Access-Hours` (in the Check-List) with the value of `Session-Timeout` (in the Return-List).

See “Allowed Access Hours” on page 99 for the format of this value (and how to enter it into a User or profile record), “SQL Statement Construction” on page 368 and “LDAP Authentication [Request] Section” on page 419 for how to use them from backend databases.

For the moment, let’s say both attributes are present in their respective lists, and that the user can be authenticated. Processing is as follows:

If the present time falls within a valid time period according to `Funk-Allowed-Access-Hours`, the server accepts the session. However, before

doing so, the server must determine the correct end time for the session. Based on this end time, the server sets an appropriate `Session-Timeout` value for the `Access-Accept` message.

- If there is a `Session-Timeout` attribute in the user's `Return-List`, Steel-Belted Radius adds this number of seconds to the present time. The result is a proposed end time.
- If the proposed end time falls within a valid time period according to `Funk-Allowed-Access-Hours`, and if this is the same time period into which the present time falls, then the original `Session-Timeout` value (from the `Return-List`) is returned.
- If the proposed end time does not fall within a valid time period according to `Funk-Allowed-Access-Hours`, or if it does, but this time period is not the one into which the present time falls, then a new `Server-Timeout` value must be calculated. The server calculates the number of seconds between the present time and the end of the valid time period into which the present time falls. This value is returned in the `Session-Timeout` attribute.
- If there is no `Session-Timeout` attribute in the user's `Return-List`, `Funk-Allowed-Access-Hours` provides the only restriction upon the user's session length. Steel-Belted Radius computes a value for `Session-Timeout` based on the number of seconds between the present time and the end of the valid time period into which the present time falls. This value is returned in the `Session-Timeout` attribute.
- If the present time does not fall within a valid time period according to `Funk-Allowed-Access-Hours`, the server rejects the session.
- If `Funk-Allowed-Access-Hours` is not present (in the `Check-List`), the server returns the `Session-Timeout` value (from the `Return-List`). If neither attribute is present, no `Session-Timeout` value is returned in the `Access-Accept` message, and the session is unlimited.

Password Protocols

During an authentication transaction, password information is transmitted between the NAS and the RADIUS server. This password information originally comes from the user, for example during PPP negotiations between a user and a NAS. Steel-Belted Radius supports four protocols for receiving the password from the NAS. Four are PPP password protocols (PAP, CHAP, MS-CHAP, and

MS-CHAP-V2). Steel-Belted Radius also supports *Extensible Authentication Protocol*.

The following table lists supported protocols according to the authentication methods with which each protocol can be used (note that some information is specific to Windows or UNIX).

Method	PAP	CHAP	MS-CHAP	MS-CHAP-V2
LDAP	Yes	Yes , if BindName is used and the password is in clear text form	Yes , if BindName is used and the password is in clear text form	Yes , can return clear-text password or MD4 hash of Unicode form of password.
		No, if Bind is used	No, if Bind is used	
Native	Yes	Yes	Yes	Yes
NT Domain Group	Yes	No	Yes , if the user is on the local domain controller	Yes , if the user is on the local domain controller
NT Domain User	Yes	No	Yes , if the user is on the local domain controller	Yes , if the user is on the local domain controller
NT Host Group	Yes	No	No	No
NT Host User	Yes	No	No	No
Windows Domain Group	Yes	No	Yes , if the user is in local or trusted domain	Yes , if the user is in local or trusted domain
Windows Domain User	Yes	No	Yes , if the user is in local or trusted domain	Yes , if the user is in local or trusted domain
Proxy RADIUS	Yes	Yes	Yes	Yes
SecurID	Yes	No	No	No
SQL	Yes	Yes , if the password is available in clear text form in the database	Yes , if the password is available in clear text form in the database	Yes , can return clear-text password or MD4 hash of Unicode form of password.
TACACS+	Yes	Yes	No	No
UNIX User	Yes	No	No	No
UNIX Group	Yes	No	No	No

PAP

Under PAP (Password Authentication Protocol), the user negotiates with the NAS “in the clear.” That is to say, no encryption is used to send the password to the NAS.

After the NAS has enough information from the user to create an Access-Request, the NAS encrypts the password (using its RADIUS authentication shared secret) before sending an Access-Request packet to Steel-Belted Radius.

Upon receiving the Access-Request, Steel-Belted Radius looks for attributes within the packet that identify the NAS that sent it. Steel-Belted Radius decrypts the password by matching this NAS with a RAS Client entry stored in its database.

Ultimately, Steel-Belted Radius has the password in clear text form for authentication.

All Steel-Belted Radius authentication methods support PAP.

CHAP

CHAP (Challenge Handshake Authentication Protocol) avoids sending passwords in clear text over any communication link.

Under CHAP, during password negotiations the NAS generates a *challenge* (a random string) and sends it to the user. The user’s PPP client creates a *digest* (the password concatenated with the challenge), encrypts the digest using one-way encryption, and sends the digest to the NAS.

The NAS sends this digest as the password in the Access-Request.

Because the encryption is one-way, Steel-Belted Radius cannot recover the password from the digest. What it can do is perform the identical digest operation using the NAS’s challenge (provided in the Access-Request packet) and its own copy of the user’s password. If the two digests match, the password is the same.

Steel-Belted Radius must be able to perform the digest operation to support CHAP. Therefore, it must have access to its own copy of the user’s password. Native User passwords are stored in the Steel-Belted Radius database. SQL or LDAP BindName authentication retrieves the password via a query to the database; the retrieved password can be used to create a digest if it is in clear text form. A TACACS+ server provides CHAP support and handles the digest operation itself after Steel-Belted Radius sends the username and password through. No other authentication method supports CHAP at this time.

MS-CHAP and MS-CHAP-V2

The two varieties of MS-CHAP (Microsoft Challenge Handshake Authentication Protocol) are Microsoft authentication protocols that, like CHAP, avoid sending passwords in clear text. Steel-Belted Radius supports both 40-bit and 128-bit MS-CHAP methods.

Steel-Belted Radius must be able to perform a digest operation similar to CHAP to support MS-CHAP. Therefore, it must have access to its own copy of the user's password. Native User passwords are stored in the Steel-Belted Radius database. SQL or LDAP BindName authentication retrieves the password via a query to the database; the retrieved password can be used to create a digest if it is in clear text form.

An NT Domain controller provides MS-CHAP support, and handles the digest operation itself after Steel-Belted Radius sends the username and password through; however, the user must be on the local domain or in a trusted domain (if Windows domain authentication is being used) for the password to be recognized.

MS-CHAP and MS-CHAP-V2 communicate users' requests to change their passwords to a RADIUS server. Steel-Belted Radius supports this feature, although it must also be supported by whatever application the user is using to log in.

MS-CHAP and MS-CHAP-V2 operate in the same way, but they use different attributes. An MS-CHAP client won't accept MS-CHAP-V2 attributes, and vice-versa; be careful to use the appropriate set of attributes.

For details about MS-CHAP and MS-CHAP-V2, see IETF RFCs 2433, 2548 and 2759.

Accounting

To understand the Steel-Belted Radius accounting sequence, you'll need an overview of RADIUS accounting messages. The following table explains the

conditions under which each type of message is issued, and the purpose of any RADIUS attributes that a message contains.

Message Conditions	Purpose of Message Attributes
<p>Accounting data is sent from client to server using an Accounting-Request message. The client manufacturer decides which types of accounting request are sent, and under which conditions. This table describes the most typical conditions.</p> <p>It is also the client's responsibility to ensure that the server receives accounting requests. Most clients retry periodically until the server responds.</p>	<p>Depending on the value of the Acct-Status-Type attribute, the message type is considered to be Start, Stop, Interim-Acct, Accounting-On, or Accounting-Off.</p>
<p>After receiving an Access-Accept from the server, the NAS completes its access negotiation with the user. The NAS then sends a Start message to the server.</p>	<p>Record connection data such as username, NAS identifier, NAS port identifier, port type, and connection start time.</p>
<p>After a connection is terminated, the NAS sends a Stop message to the server.</p>	<p>Record statistics regarding the connection. One message contains the final value of every statistic that this NAS is capable of recording about this type of connection.</p>
<p>At intervals of approximately every 6 minutes, the NAS sends an Interim-Acct message to the server.</p>	<p>Record a "snapshot" of statistics regarding the connection. One message contains the current value of every statistic that this NAS is capable of recording about this type of connection.</p>
<p>Every time a client device comes online, whether after a crash or after an orderly shutdown, it sends an Accounting-On message to the server.</p>	<p>Identify the device that is going online and clear all session information.</p>
<p>Every time a client device experiences an orderly shutdown, before completing its shutdown sequence it sends an Accounting-Off message to the server.</p>	<p>Identify the device that is going offline and clear all session information.</p>
<p>Upon receipt of an Accounting-Request message, the server sends an Accounting-Response.</p>	<p>Complete the request/response cycle.</p>

Accounting Sequence

A NAS can issue an Accounting-Request whenever it chooses, for example upon establishing a successful connection. Each time an Accounting-Request message arrives at the Steel-Belted Radius server, an accounting transaction begins. During

this transaction, the server handles the message by examining the Acct-Status-Type and other attributes within the message, and taking the appropriate action.

Comma-Delimited Log Files

When the Steel-Belted Radius accounting log is enabled, all of the RADIUS accounting attributes that the server receives are reformatted and logged to a Comma Separated Value (CSV) text file, which is easily imported into spreadsheets and database programs for report generation and billing.

Proxy RADIUS Accounting

Steel-Belted Radius can relay an Accounting-Request to some other RADIUS server, which records the data according to its own, locally-configured RADIUS accounting options. (You have the option of specifying that the data also be recorded locally on the Steel-Belted Radius server.) The set of conventions for relaying packets between cooperating RADIUS servers is known as *Proxy RADIUS*, and is well-defined in the RADIUS standard.

Note: Steel-Belted Radius offers a wide range of powerful accounting options which build upon Proxy RADIUS accounting and realms. See “Configuring a Proxy RADIUS Realm” on page 257.

External Accounting

External accounting methods permit Steel-Belted Radius to record accounting data by using configuration files that tell it how to (1) communicate with an external database and (2) insert accounting data into that database. The only external accounting method currently supported is SQL.

See “SQL Accounting” on page 388.

Tunneled Accounting

During authentication, a user is typically identified by attributes such as User-Name (in the authentication request) and Class (in the authentication accept response). Standard RADIUS accounting requests typically include these attributes in messages flagging Start, Interim, and Stop events so that the user’s identity can be recorded for accounting and auditing purposes.

When an organization uses a tunneled authentication protocol such as EAP/TTLS or EAP/PEAP, the identity of a user requesting authentication may be concealed from the NAS; the User-Name attribute carried by the outer authentication protocol is typically a non-unique value such as ‘anonymous.’ As a result, the outer User-Name value included in accounting requests may not be sufficient to determine a user’s

identity, and Class attributes provided by an authentication server cannot be included in cleartext in an outer Access-Accept message because they might contain clues about the user's identity, thereby defeating the identity-hiding feature of the tunneled protocol.

Tunneled accounting allows Steel-Belted Radius to pass user identity information to accounting processes without exposing user identities to a NAS or AP that should not see them. When tunneled accounting is enabled, RADIUS attributes are encrypted and encapsulated within one or more¹ class attributes.

- 1 The Steel-Belted Radius server acting as the tunnel endpoint for EAP/TTLS or EAP/PEAP encrypts a user's inner User-Name and Class attributes when it authenticates the user.
- 2 The server returns the encrypted information to the NAS or AP encapsulated in a Class attribute in the outer Access-Accept message. The NAS or AP associates this encapsulated identity attribute with the user, and echoes the encapsulated identity attribute whenever it generates an accounting request for the user.
- 3 When the Steel-Belted Radius server receives an accounting request from a NAS or Access Point, the server scans the request for an encapsulated identity attribute.
- 4 If the server finds an encapsulated identity attribute, it de-encapsulates and decrypts the attributes to reconstitute the original inner User-Name and Class attributes.
- 5 The server substitutes the decrypted attributes for the ones returned from the NAS or AP.
- 6 The server processes the accounting request locally or forwards the accounting request via proxy to its intended target.

To implement tunneled accounting, you must configure the classmap.ini file to specify how attributes should be presented, and you must configure the spi.ini file to specify the keys that are used to encrypt and decrypt users' identity information. The classmap.ini file is described in "classmap.ini File" on page 193. The spi.ini file is described in "spi.ini File" on page 229.

For an overview of how EAP/TTLS and EAP/PEAP work, refer to "EAP Types" on page 305.

1. If the information for a Class attribute exceeds the attribute payload size (253 octets), Steel-Belted Radius returns more than one Class attribute for a user.

Directed Accounting

The directed accounting feature allows you to map an incoming accounting request to one or more accounting methods, based on routing information found in the request packet. Among the options available with directed accounting is that of establishing an accounting log file that is distinct from the Steel-Belted Radius accounting log file in the server directory, and that contains entries from only those accounting requests that were specifically directed to the realm.

See “Configuring a Directed Realm” on page 263.

Accounting Spooling

Accounting Spooling can improve both proxy accounting performance and reliability.

When spooling is enabled for a realm, Steel-Belted Radius immediately acknowledges all accounting requests for that realm to the NAS. Meanwhile, it spools accounting requests to a file while a separate thread unspools requests and sends them to the server responsible for the realm. If the server is unavailable, Steel-Belted Radius retries at regular intervals until the proxy server acknowledges the request. Even if Steel-Belted Radius restarts, all spooled requests are retained until they are completed.

This scheme has the following benefits:

- The NAS always gets an immediate ACK (acknowledgement response) for accounting requests.
- Accounting data is never lost if it is sent to a Steel-Belted Radius server with spooling enabled.

There is a separate and independent spooler for each realm for which Proxy Spooling is configured, such as an ISP customer or a remote centralized accounting target. When an accounting request is received for a realm implementing Proxy Spooling, it is written to a file in the target directory and a request is prepared, followed by an acknowledgement returned to the client. The file is then read by the unspooling thread and the prepared request proxied.

Targets, fast-fail, round-robin, and other extended proxy features operate normally, but unspooling continues to retry sending a request until it is successfully acknowledged. Since each spooler is independent, one unresponsive realm does not affect the delivery of spooled requests to other realms.

The Acct-Delay-Time attribute in a request is updated or added as necessary if there is a delay between the spooling and the forwarding of the request.

When the pool file's rollover interval expires or the file size exceeds the rollover size limit, the current pool file is closed for writing and a new one created. Files are named in the format, *yyyymmdd_hhmm_ssss.psf*, where *yyyy* is the year, *mm* is the month, *dd* is the day, *hh* is the hour, *mm* is the minute, and *ssss* is a sequence number. The configuration of the rollover settings enables spooling to be optimized according to the characteristics of the operating environment.

When Steel-Belted Radius is shutdown, unspooling continues for the configured `ShutdownDelay` time until all spooled packets are sent. If the destination server is down at the time of shutdown, however, unspooling terminates immediately. After startup, unspooling continues from the beginning of the oldest spool file.

See "Proxy RADIUS [SpooledAccounting] Section" on page 290.

Sessions List (Current Users Display)

In addition to simply recording RADIUS accounting data, Steel-Belted Radius also processes the data to gather its own statistics, including a real-time snapshot of currently active connections called the Sessions List (also called the Current Users display). You can view this display at any time by clicking a button on the Administrator program's Statistics dialog. For every active connection, a line is displayed identifying the user, the NAS, the port number, the assigned IP address, and other information.

Each server has its own Current Users display. Therefore, when you view this display, it reflects only the activity on the Steel-Belted Radius server that you've currently selected for administration (using the Servers dialog). The Current Users display on a specific server reflects the activity across your entire RADIUS configuration only if (1) all clients in your configuration support RADIUS accounting, and (2) all clients are configured to send accounting messages to the same server (the one you're viewing).

See "Sessions List" on page 158.

Attributes

You'll work with RADIUS attributes while setting up Users, profiles, and RAS Clients on the Steel-Belted Radius server. You won't need to memorize the RADIUS standard or work in hexadecimal (that is, "packet") format to do this. The Steel-Belted Radius Administrator program allows you to select RADIUS attributes by name from a predetermined list. For each attribute, the Administrator program prompts you to enter values using familiar data types such as string, integer, telephone number, or network address.

This section provides all of the background information you need to work with attributes on the Steel-Belted Radius server.

Dictionaries

Steel-Belted Radius uses files called *dictionaries* to store lists of RADIUS attributes. The main Steel-Belted Radius dictionary file `radius.dct` lists attributes defined by the RADIUS standard. The `radius.dct` file resides in the same directory as the Steel-Belted Radius service (usually `C:\RADIUS\Service` on Windows computers or `Radius_Home\` on UNIX computers).

Vendor-Specific Attributes

In addition to the standard attributes, many NASs use additional, Vendor-Specific Attributes (VSAs) to complete a connection. Steel-Belted Radius supports a large number of specific NAS devices by providing vendor-specific, proprietary dictionary files. These files also reside in the server directory and use the filename extension `.dct`.

Make/model Field

During Steel-Belted Radius configuration, when you make a selection in the RAS Client **Make/model** field, you are telling the server which dictionary file contains the VSAs for this client device. Thereafter, whenever the server receives a RADIUS packet from this client device, it can consult this dictionary file for any non-standard attributes that it encounters in the packet. Standard RADIUS attributes are always defined by the `radius.dct` file. If you are in doubt as to the Make/model that you should choose for a RAS Client, it's a safe bet to choose the default option, **- Standard Radius -**.

For the most part, the selections currently available in the **Make/model** field are devices whose vendors have worked with Funk Software to provide up-to-date attribute dictionaries. Documentation for these vendors and their products is available online by clicking on the **Vendor info** button on the RAS Clients dialog.

See "RAS Clients Dialog" on page 87.

If you are using a computer running Windows, you can also access product information by selecting **Help > Vendor Info** from the Administrator menu bar.

Updating Attribute Information

If you receive news from your NAS vendor about a new product, a new attribute, or a new value for an attribute, you can add this information to your Steel-Belted

Radius configuration. You can edit the dictionary file for that vendor to add new attributes or attribute values, or you can create a new vendor-specific dictionary file that contains new attributes and values.

For detailed instructions, see “Dictionary Files” on page 239.

User Attribute Lists

Each User entry in the Steel-Belted Radius database provides the information necessary for the server to try to authenticate a connection request using a specific authentication method. When you view a User entry using the Administrator program, this method is identified in the **User type** field.

There can be more to authentication than a simple username/password pair. If you want, you can control authentication at a fine level of detail. The Check-List, Return-List, or profile fields in the User entry in the database provide powerful tools for the authentication and authorization of users. These fields tell the server how to handle RADIUS attributes while authenticating a connection request and can be used to configure the authorization of the session.

Note: All of these fields are optional.

Check-List Attributes

The *Check-List* is a list of attributes that must accompany the request for connection and thus could be considered “authentication requirements.” The NAS must send attributes that match the Check-List that is “on file” in a User entry; otherwise, Steel-Belted Radius rejects the user even if the user’s name and password are valid.

By including appropriate attributes in the Check-List, a variety of rules could be enforced. For example, only specific users might be permitted to use ISDN or dial-in connections to a particular NAS, or Caller ID might be used to validate a user against a list of legal originating telephone numbers.

A Check-List is created by selecting attributes from a list of all RADIUS attributes known to the Steel-Belted Radius server. This list can include a variety of vendor-specific attributes.

During authentication, Steel-Belted Radius “filters” the Check-List based on the dictionary for the Make/model of the specific RAS Client that sent the authentication request. The server ignores any Check-List attribute that is not valid for this device.

Return-List Attributes

The *Return-List* is a list of attributes that Steel-Belted Radius must return to the NAS after authentication succeeds. The Return-List usually provides additional parameters that the NAS needs to complete the connection, typically as part of PPP negotiations. They can thus be considered to be “authorization configuration parameters.”

By including appropriate attributes in the Return-List, a variety of connection policies could be applied. Specific users could be assigned particular IP addresses or IPX network numbers, IP header compression could be turned on or off, or a time limit could be assigned to the connection.

A Return-List is created by selecting attributes from a list of all RADIUS attributes known to the Steel-Belted Radius server. This list can include a variety of vendor-specific attributes.

During authentication, Steel-Belted Radius “filters” the Return-List based on the dictionary for the Make/model of the specific RAS Client that sent the authentication request. The server omits any Return-List attribute that is not valid for this device.

Attribute Values

The value of each RADIUS attribute has a well-defined data type, which can be numeric, string, IP or IPX address, time, or hexadecimal.

For example, `Callback-Number` is of type `string` and contains a telephone number. `NAS-Port-Type` is an item from a list, and can be `Sync`, `Async`, and so forth.

Multi-valued Attributes

Attributes can be single- or multi-valued; in other words, some attributes might appear at most once in the Check-List or Return-List, while others might appear several times.

If an attribute appears more than once in the Check-List, this means that any one of the values is valid. For example, you can set up the Check-List to include both `Sync` and `Async` values for attribute `NAS-Port-Type`. This means that the user can dial into a Sync port or an Async port, but not one of the ISDN ports.

If an attribute appears more than once in the Return-List, this results in each value of the attribute being sent as part of the response packet. For example, to enable both IP and IPX header compression for a user, the `Framed-Compression` attribute should appear twice in the Return-List: once with the value

VJ-TCP-IP-header-compression and once with the value IPX-header-compression.

Orderable Attributes

Certain multi-valued Return-List attributes are also orderable; that is, the attribute can appear more than once in a RADIUS response, and the order in which the attributes appear is important.

For example, the Reply-Message attribute allows text messages to be sent back to the user for display. A multi-line message is sent by including this attribute multiple times in the Return-List, with each line of the message in its proper sequence.

System Assigned Values

Some attributes do not allow the administrator to set a value. Steel-Belted Radius retrieves the appropriate value for this attribute when it is needed.

Echo Property

Using the echo property, you can force an attribute from the RADIUS request to be echoed in the RADIUS response.

Example: You add Callback-Number to the Return-List and select the **echo** checkbox. Steel-Belted Radius takes the value of the Callback-Number it receives in the RADIUS request and echoes it back to the client in the RADIUS response; if it receives no Callback-Number, it echoes nothing.

You enter Callback-Number one or more times into the Check-List. This indicates that one of the callback numbers you supplied must be present in the RADIUS request, and that number should be echoed in the RADIUS response.

Default Values

By selecting **default** for any Check-List attribute, you indicate that if the RADIUS request does not include this attribute, the request should not be rejected. Instead, the value supplied as the default should be used as if it were received as part of the request.

One use for default values is to require that an attribute in a RADIUS request must have one of several values, or must not be present at all.

Another use would be to provide a default value for an attribute in conjunction with the echo property in the Return-List. If an attribute appears once in the Check-List

marked as **default**, and the same attribute appears in the Return-List marked as **echo**, this means the following:

- If the attribute does appear in the RADIUS request, the server echoes it in the RADIUS response.
- If the attribute does not appear in the RADIUS request, the server echoes the default value (from the Check-List) in the response.

If you add multiple values of the same attribute to the Check-List, only one of them can be marked as **default**.

Suppose, for example, you add several Callback-Number values to the Check-List and mark one of them as default. Also, you add `Callback-Number` to the Return-List and specify it as **echo**. Here's what happens:

- If a Callback-Number value is present in the RADIUS request, it must match one of the Check-List values or the user is rejected.
- If it does match, the user is accepted and the value supplied is echoed in the RADIUS response.
- If no `Callback-Number` is supplied in the request, the user is accepted and the default value is echoed in the response.
- Other Check-List attributes are used to provide configuration for the user, such as time-of-day and concurrent-login-limit information.

Wildcard Support

Steel-Belted Radius supports wildcards ('?' and '*') for string-type attributes in checklist items and for IP addresses using a network number.

To allow backward compatibility with checklist items that treat the string literally, a string containing wildcards must be prefixed with a caret ('^'). When the caret is present, the remainder of the string is parsed using escape rules.

A '?' matches any character and an '*' matches the remainder of the string (but can appear only at the end of a string). Wildcard characters can be treated as literals by using escape codes (for example, '\?'). The following non-ASCII characters can also be present in the wildcard string:

Code	Meaning
\a	BEL
\b	BS
\f	FF

Code	Meaning
\n	LF
\r	CR
\t	HT
\v	VT
\\	Backward backslash
*	Literal '*' (not wildcard)
\?	Literal '?' (not wildcard)
\xnn	Where <i>nn</i> is a hexadecimal value
\nnn	Where <i>nnn</i> is a decimal value

A ‘\’ followed by any other character represents that character’s value.

The following is a wildcard example for string type attributes:

```
Called-Station-ID = ^800*
```

where `Called-Station-ID` indicates any 800 number.

The following is a wildcard example for IP Addresses:

```
NAS-IP-Address = 199.100.10.0
```

where `NAS-IP-Address` indicates any IP address on the 199.100.10.0 network.

Attribute Filtering

You can filter specific RADIUS attribute/value pairs into and out of RADIUS packets as they travel to and from directed realms and Proxy RADIUS realms. This can be useful, for example, if there is data in the packets that is needed for routing, but not for authentication or accounting.

To configure an attribute filter, see “filter.ini File” on page 202.

Profiles

Steel-Belted Radius lets you define default templates of Check-List/Return-List pairs called *profiles*. A profile provides specific attributes for one or both lists. You can define as many profiles as you want. This feature provides a powerful means of managing and configuring accounts.

See “Profiles Dialog” on page 111.

When you edit a User entry, you can select a profile. When you do, Steel-Belted Radius assigns this profile to the User. That is, this profile's Check-List and Return-List attributes become the default settings for the User entry. This saves time (and the risk of typing errors!) compared to editing the lists individually. Profiles thus provide a management solution that is scalable and efficient in terms of time and storage space.

After you assign a profile to a User entry, you are free to modify the new entries on the User's Check-List and Return-List. All of the changes you make are local variations that apply only to this User entry; they do not affect the profile itself. Assigning a profile and then overriding individual attributes is a convenient way to leverage Steel-Belted Radius's features to your advantage.

See "Editing User Settings" on page 96.

To change attributes settings across many users immediately, edit the profile that you've assigned to these users; the changes you make to the profile are automatically reflected in each user's Check-List and Return-List.

Resolving profile and User Attributes

If there are user-specific attributes stored in an external database, the Steel-Belted Radius server determines the final set of attributes for a user by merging the attributes stored in the native database with those retrieved from the external database. This calculation is performed as follows:

- 1 The attributes from the profile (or Alias user) assigned to the user are first retrieved.
- 2 These attributes are then merged with the user-specific modifications to the attributes in the following manner:
 - If the attribute is multi-valued, then the attribute(s) retrieved from the external database is added to the overall list of attributes.
 - If the attribute is single-valued, then the attribute(s) retrieved from the external database replaces any attribute of the same name in the profile or associated with the alias.
 - If the attribute is orderable, then the attribute(s) retrieved from the external database replaces any orderable attribute of the same name in the profile or associated with the alias.

Request Routing

When a RADIUS authentication or RADIUS accounting request arrives at the Steel-Belted Radius server, the server examines the attributes in the request to determine its correct destination. By this we mean that Steel-Belted Radius must match the request with the service that can best respond to it; for example:

- RADIUS authentication or accounting
- Proxy RADIUS authentication or accounting
- Tunnel authentication
- Directed authentication or accounting

We call this matching process *request routing*. The information used to route the request can be found in the User-Name attribute (in which routing information is supplied as a prefix or suffix to the user's account name), the Called-Station-Id attribute (DNIS), or the specific attribute(s) of your choice.

Steel-Belted Radius first checks the User-Name and Called-Station-Id attributes, then it checks the attributes you've mapped to realms. It uses the first routing destination it finds.

The following sections describe request routing in detail.

User-Names with a Single Delimiter

An incoming User-Name string can be “decorated” with a single delimiter separating the user's name from a destination name. A User-Name decorated in this manner might indicate a Proxy RADIUS realm, a directed realm, a Tunnel, or a Proxy entry that is not a member of any realm.

Note: To prevent unexpected routing results, you must ensure that the name of every realm, Tunnel, and Proxy entry is unique across your entire Steel-Belted Radius configuration.

Steel-Belted Radius determines the destination as follows.

User-Names with a Single Tunnel Delimiter

If the delimiter matches the currently configured delimiter for Tunnels, and if the current name-parsing convention for Tunnels is `suffix`, the User-Name is understood to be:

User<SuffixDelimiter>TunnelName

or, if the current name parsing convention for Tunnels is `prefix`, the User-Name is understood to be:

TunnelName<*PrefixDelimiter*>*User*

where:

User is the name of the dial-in user;

TunnelName identifies the destination; and

<*SuffixDelimiter*> or <*PrefixDelimiter*> is a delimiter character such as '@', '/' or '!'.

If a Tunnel entry is found that matches the *TunnelName*, and the request is for authentication, Steel-Belted Radius proceeds with tunnel authentication.

Note: Tunnel delimiters are defined in the Configuration dialog. You can use either the prefix or the suffix naming convention for tunnels, but not both. You can also choose the tunnel delimiter character ('@', '/', and so forth). The conventions you define in the Configuration dialog apply to all tunnels defined on the server.

User-Names with a Single Realm Delimiter

If the User-Name contains a single delimiter that matches the currently configured suffix delimiter for realm destinations, the User-Name is understood to be:

User<*SuffixDelimiter*>*RealmName*

or, if the User-Name contains a single delimiter that matches the currently configured prefix delimiter for realm destinations, the User-Name is understood to be:

RealmName<*PrefixDelimiter*>*User*

where:

User is the name of the dial-in user

RealmName identifies the destination

<*SuffixDelimiter*> or <*PrefixDelimiter*> is a delimiter character such as '@', '/' or '!'.

Steel-Belted Radius attempts to find a destination that matches *RealmName*. A match might be found in one of four places:

- The [Self] section of the radius.ini file. In this case, Steel-Belted Radius services the request locally.
- The [Directed] section of the proxy.ini file. If a match is found here, Steel-Belted Radius routes the request to a specific authentication or

accounting method on the local server, according to the rules in the corresponding *RealmName.dir* file.

- The [Realms] section of the *proxy.ini* file. If a match is found here, Steel-Belted Radius routes the request to the Proxy RADIUS realm called *RealmName* according to the rules in the corresponding *RealmName.pro* file. See “Target Selection Within a Realm” on page 68.
- A Proxy entry in the Steel-Belted Radius database. If a match is found here, Steel-Belted Radius uses the information in this Proxy entry (IP address, UDP port, shared secret) to forward the RADIUS request.

Note: *Realm delimiters and naming conventions are defined in the proxy.ini file. You can define different delimiters for prefixes and suffixes. The conventions you define in proxy.ini apply to all types of realm defined on the server (both Proxy RADIUS realms and directed realms).*

User-Names with Multiple Suffix Delimiters

If the User-Name contains multiple realm delimiters such as:

User<Delimiter>RealmName<Delimiter>RealmName<Delimiter>RealmName

and the delimiter character matches the current *RealmSuffix* setting in the [Configuration] section of *proxy.ini*, the name parsing strategy is as follows:

- 1 Steel-Belted Radius finds the leftmost *RealmName* in the User-Name that is also listed in the [Self] section of its *radius.ini* configuration file.
- 2 If a matching *RealmName* was found in Step 1, and there is no other *RealmName* to the left of it, then Steel-Belted Radius services the request locally, without forwarding.
- 3 If a matching *RealmName* was found in Step 1, but there is another *RealmName* to the left of it, then Steel-Belted Radius routes the request to the *RealmName* listed immediately to the left of the matching *RealmName*. The routing is controlled by the corresponding *RealmName.pro* or *RealmName.dir* file.
- 4 If no *RealmName* was selected in Steps 1, 2, or 3, then Steel-Belted Radius routes the request to the rightmost *RealmName* in the User-Name. The routing is controlled by the corresponding *RealmName.pro* or *RealmName.dir* file.

According to these rules, if the realm suffix *Delimiter* character is ‘@’, and the User-Name value matches realm suffix naming conventions, and the [Self] section

of `radius.ini` lists one realm called `bigserver`, then incoming User-Name values would be parsed as follows:

A request for...	Would be...
<code>fred@bignet@bigserver</code>	Routed to the realm called <code>bignet</code> .
<code>fred@bignet@bigserver@smallnet</code>	Routed to the realm called <code>bignet</code> .
<code>fred@bignet@smallnet</code>	Routed to the realm called <code>smallnet</code> .
<code>fred@bigserver@bignet</code>	Handled locally on <code>bigserver</code> .

User-Names with Multiple Prefix Delimiters

If the User-Name contains multiple realm delimiters such as:

RealmName<*Delimiter*>*RealmName*<*Delimiter*>*RealmName*<*Delimiter*>*User*

and the *Delimiter* character matches the current `RealmPrefix` setting in the [Configuration] section of `proxy.ini`, the name parsing strategy is the reverse of the suffix strategy described above. In detail:

- 1 Steel-Belted Radius finds the rightmost `RealmName` in the User-Name that is also listed in the [Self] section of its `radius.ini` configuration file.
- 2 If a matching `RealmName` was found in Step 1, and there is no other `RealmName` to the right of it, then Steel-Belted Radius services the request locally, without forwarding.
- 3 If a matching `RealmName` was found in Step 1, but there is another `RealmName` to the right of it, then Steel-Belted Radius routes the request to the `RealmName` listed immediately to the right of the matching `RealmName`. The routing is controlled by the corresponding `RealmName.pro` or `RealmName.dir` file.
- 4 If no `RealmName` was selected in Steps 1, 2, or 3, then Steel-Belted Radius routes the request to the leftmost `RealmName` in the User-Name. The routing is controlled by the corresponding `RealmName.pro` or `RealmName.dir` file.

According to these rules, if the realm prefix *Delimiter* character is `!`, and the User-Name matches realm prefix naming conventions, and the [Self] section of `radius.ini` lists one realm called `bigserver`, then incoming User-Name values would be parsed as follows:

A request for...	Would be...
<code>superserver!bignet!fred</code>	Routed to the realm called <code>bignet</code> .
<code>smallnet!bigserver!bignet!fred</code>	Routed to the realm called <code>bignet</code> .

A request for...	Would be...
smallnet!bignet!fred	Routed to the realm called smallnet.
bignet!bigserver!fred	Handled locally on bigserver.

Request Routing by DNIS

If the Called-Station-Id attribute is found in the RADIUS request, the request can be routed based on DNIS (Dialed Number Information Services). Steel-Belted Radius checks its administration database and server configuration files for a DNIS string that matches the value of the incoming Called-Station-Id attribute. If found, the matching string might be in one of three places:

- A Tunnel entry’s Called Station Id list. If a match is found here, and the request is for authentication, Steel-Belted Radius performs tunnel authentication.
See “Tunnel Authentication Sequence” on page 72.
- The [Called-Station-ID] section of a *RealmName.dir* file. If a match is found here, Steel-Belted Radius handles the request locally using the authentication and/or accounting methods identified in the *RealmName.dir* file.
See “Configuring a Directed Realm” on page 263.
- The [Called-Station-ID] section of a *RealmName.pro* file. If a match is found here, Steel-Belted Radius routes the request to the Proxy RADIUS realm called *RealmName* using the rules defined in the *RealmName.pro* file.
See “Target Selection Within a Realm” on page 68.

Note: We strongly suggest that you use DNIS strings that are unique across all Tunnel entries, all *RealmName.dir* files, and all *RealmName.pro* files. If you duplicate a DNIS string anywhere in your Steel-Belted Radius configuration, the request routing results might be unexpected.

Request Routing by Any Attribute

You can map the presence or absence of any attribute or attribute/value pair in an incoming packet to a specific realm, by providing an [AuthAttributeMap] or [AcctAttributeMap] section in the proxy.ini configuration file.

You can route all of the packets for a session to a realm based on attributes in the Access-Request (the [AuthAttributeMap] section), or you can route the session’s accounting packets to a different realm, based on attributes found in these packets (the [AcctAttributeMap] section).

Attribute mapping can be used for Proxy RADIUS realms and for directed realms. You cannot use this feature when forwarding packets to a Proxy target that is not a member of a realm.

See “proxy.ini [AttributeMap] Sections” on page 269.

Local Services

If the RADIUS request does not contain routing information (or at least, it does not contain any routing information that Steel-Belted Radius has been configured to recognize), it is processed locally on the Steel-Belted Radius server. Authentication follows the Authentication Methods list in the server’s Configuration dialog. No User-Name parsing is performed; the entire string is understood to be the user’s name. Accounting is controlled by the server’s main account.ini file and (for external database accounting) .acc file.

Control Over Routing Methods

The rules for determining the destination of a request are applied in the following order by default:

- 1 Apply Suffix delimiter rules
- 2 Apply Prefix delimiter rules
- 3 Apply DNIS rules
- 4 Apply Attribute Mapping rules

You can now, however, specify which of these methods you want applied in the routing of requests and the order in which they are applied.

For details on configuration, see “proxy.ini [Processing] Section” on page 275.

Proxy RADIUS

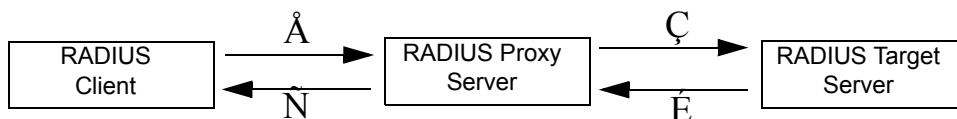
Steel-Belted Radius can forward a RADIUS request to another server for processing and relay the other server’s result back to its client. We say that Steel-Belted Radius is acting as a *proxy* for the *target* server, and that Steel-Belted Radius is *proxy-forwarding* the request to the target server.

Steel-Belted Radius fully supports Proxy RADIUS, in that every computer running it can act as either proxy or target for either authentication or accounting messages.

Proxy RADIUS Authentication

RADIUS authentication messages are proxy-forwarded as follows:

- 1 A RADIUS client sends an authentication request to a proxy RADIUS server.
- 2 The proxy RADIUS server forwards the message to a *target* RADIUS server.
- 3 The target RADIUS server performs the authentication services indicated by the message, then returns a response message to the proxy RADIUS server.
- 4 The proxy RADIUS server relays the response message to the RADIUS client.



Proxy RADIUS Accounting

RADIUS accounting messages are proxy-forwarded as follows:

- 1 A RADIUS server receives an accounting request.
- 2 Depending on its configuration, the RADIUS server forwards the accounting message to a target server or records accounting attributes locally on the proxy server or does both.
- 3 If the proxy server does not receive an acknowledgement of the forwarded packet, it re-sends periodically according to its retry policy.

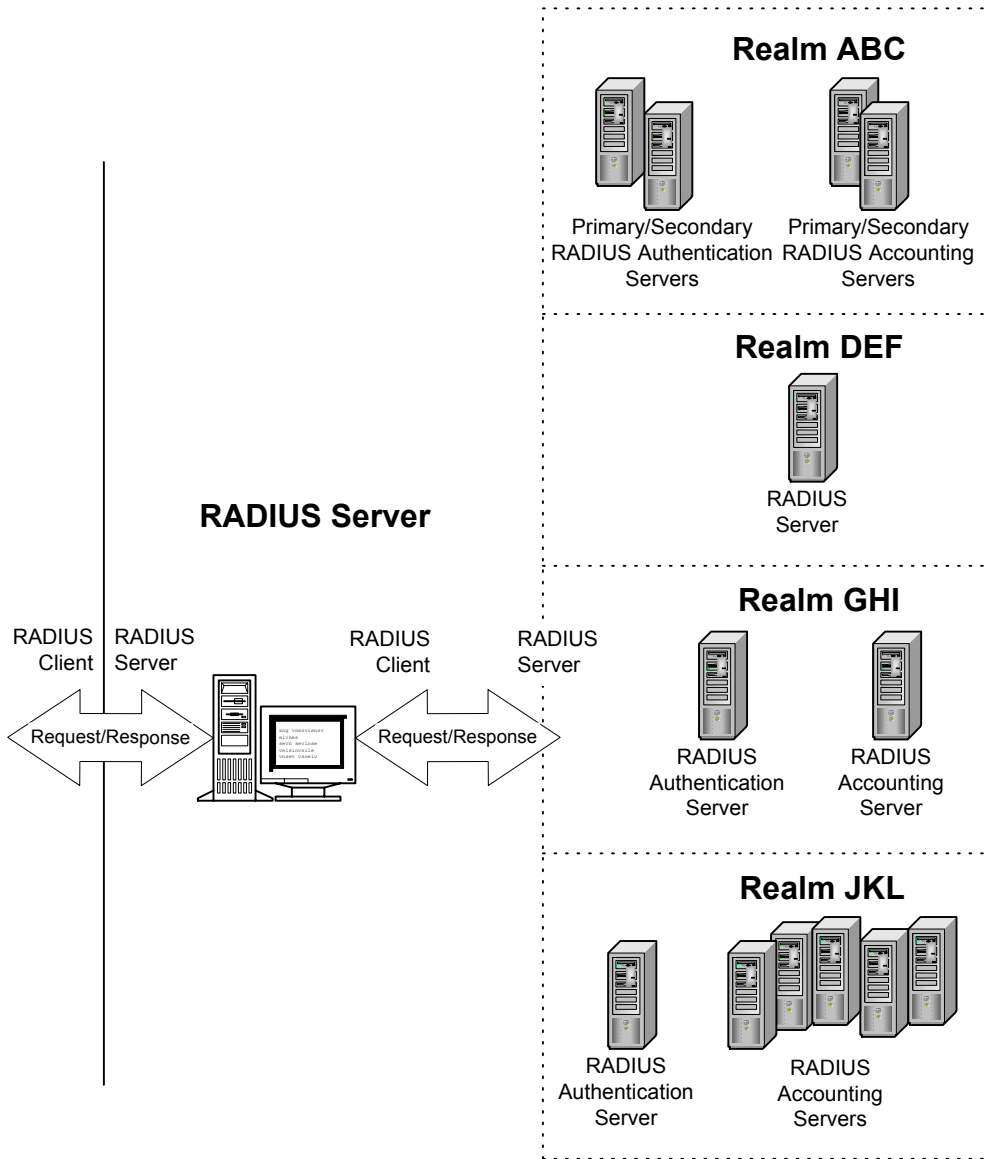
Proxy RADIUS Realms

Proxy RADIUS realms are pools of RADIUS servers to which Steel-Belted Radius can forward RADIUS requests. Proxy RADIUS realms can be configured to support workload distribution, redundancy, fault tolerance, retry policies, primary-secondary server roles, and separation of authentication and accounting responsibilities by server.

A carrier with Steel-Belted Radius installed on its LAN might create a realm that consists of all the RADIUS servers owned by a particular service provider. In this case, the RADIUS servers are already owned and maintained by the service provider; the realm simply organizes the routing of RADIUS packets from Steel-Belted Radius to these servers.

The carrier might define several such realms, one for each service provider that employs its services. If a service provider's network is extremely large, a carrier might decide to use several realms to represent a single service provider. For each of these realms, it is possible to define an independent set of conventions for storing, forwarding, routing, and filtering the RADIUS requests that enter the Steel-Belted Radius server.

See "Configuring a Proxy RADIUS Realm" on page 257.



RADIUS server and Realms

Target Selection Within a Realm

For proxy RADIUS realms, after the destination realm is identified, Steel-Belted Radius must next select a target within the realm. Target selection depends upon a number of factors, all of which you can set up in advance by editing the realm

configuration files on the Steel-Belted Radius server: proxy.ini, radius.ini, filter.ini, and one *RealmName*.pro file per realm.

See “Proxy RADIUS Target Selection Rules” on page 286.

After the target is selected, Steel-Belted Radius matches the target name with a Proxy entry in its database. Using the data in this entry (IP address, UDP port, shared secret) Steel-Belted Radius establishes a connection between itself and the target, and proxy-forwards the RADIUS request. Note that you can configure the realm so that all realm routing information and delimiters are stripped from the User-Name before forwarding.

The target processes the request as it normally would for RADIUS authentication or accounting. In the case of authentication, Steel-Belted Radius waits for a response from the target, then relays this response to its RADIUS client.

Message-Authenticator Support

The Message-Authenticator attribute enables Steel-Belted Radius to determine whether the packet received is from an actual proxy server. It might also sign the forward request.

Steel-Belted Radius can be configured to use the Message-Authenticator attribute when forwarding packets using proxy RADIUS. It can also be configured to validate or ignore the Message-Authenticator if included in the packets received.

See “Proxy RADIUS [Auth] Section” on page 279.

Proxy Fast-Fail

During proxy forwarding, Steel-Belted Radius acts as the RADIUS client of another RADIUS server. Since RADIUS clients take responsibility for delivering RADIUS packets, all of them have a “retry policy” that determines how often and for how long they continue to try to deliver a packet until they receive the response that they expect from the RADIUS server. This includes the Steel-Belted Radius server when it acts as the RADIUS client of a Proxy RADIUS target server.

Steel-Belted Radius provides a “fast-fail” option for Proxy RADIUS realms. This feature saves the Steel-Belted Radius server from continuing to send packets to a target server that appears to be down temporarily.

Let’s say that Steel-Belted Radius is sending a packet to a target and it is not getting a response within the amount of time it expects. It keeps trying periodically to send the packet until it has used up the number of tries in its retry policy. If it still hasn’t received a response from the target at that point, Steel-Belted Radius removes the target from the active list and places it on the fast-fail list.

Each time a request from a realm is received, Steel-Belted Radius sends a probing request to all fast-fail entries for this realm. No response is expected or required from the probes. No retry policy is followed. If a response to a probe is received, that target is removed from the fast-fail list. When the fast-fail timer expires for a target, it is placed back on the active list.

We strongly recommend that you specify a [FastFail] section in each Proxy RADIUS realm configuration (.pro) file. The [FastFail] section permits you to fine-tune retry policies for individual realms, or for specific targets within realms. Any [FastFail] settings that you supply in a .pro file override the current `ProxyFastFail` setting.

See “Proxy RADIUS [FastFail] Section” on page 289.

The `radius.ini` file offers a `ProxyFastFail` setting for single-target Proxy entries that are not a member of any realm. `ProxyFastFail` has an integer value, usually 1800. If a target remains on the fast-fail list longer than this number of seconds, it is automatically removed from the fast-fail list. If conditions warrant, a target may be returned to the fast-fail list at any time.

See “radius.ini [Configuration] Section” on page 212.

Static Proxy Accounting

“Static” proxy accounting allows you to send duplicate copies of certain types of accounting message to Proxy RADIUS realms, in addition to the normal routing of the original accounting message. The number of duplicates is not limited.

Static proxy accounting doesn’t prevent the request from being dynamically routed for RADIUS accounting services based on User-Name decoration, DNIS number, or attribute mapping, nor does it prevent local logging or other accounting methods from occurring. Additionally, if static proxy-forwarding fails (due to a lack of response from the target) this does not prevent the original RADIUS accounting request from being acknowledged.

An important function of static proxy accounting is to ensure that `Accounting-On` and `Accounting-Off` messages can be routed to realms. A NAS (RADIUS client) normally issues these standard accounting messages to its RADIUS server when it goes online (`Accounting-On`) and offline (`Accounting-Off`). In either case, it is understood that all connections previously made by this NAS are now invalid, and the RADIUS server is now entitled to free up any resources that it allocated to those sessions.

Static proxy accounting is necessary to deliver `Accounting-On` and `Accounting-Off` messages to realms, because these messages do not contain the `User-Name` or `Called-Station-Id` attributes that Steel-Belted Radius would normally use to route packets to realms.

Let's say the original Access-Request, an authentication message, was used to determine the realm destination for both authentication and accounting for a particular session. The attribute used to route the Access-Request may have been the `User-Name`, the `Called-Station-Id`, or any other RADIUS attribute in the Access-Request, depending on how you've configured request routing for authentication messages.

To use an LDAP directory or SQL Server to determine the routing of an authentication request, see "Routed Proxy" on page 461.

Accounting packets for this same session can be matched with the realm destination only if the server knows which session is involved (as it does in `Start`, `Stop`, and `Interim` messages). The `Accounting-On` and `Accounting-Off` messages are independent of specific sessions; therefore it is impossible to route them to realms without additional information.

By setting up static proxy accounting, and listing all realms as targets for `Accounting-On` and `Accounting-Off` messages, you can at least ensure that network information (such as NAS status) is sent to everyone that might require it.

See "proxy.ini [StaticAcct] Section" on page 276.

Proxy AutoStop Feature

A user session can be removed from the current sessions table of the Steel-Belted Radius server in ways other than the usual `Accounting-Stop` message from the NAS:

- An `Accounting-On` or `Accounting-Off` message received from the NAS causes all sessions originating from this NAS to be purged, as this signals either that the NAS has been restarted, or is now going down, respectively.
- The administrator can remove users via the LCI.
- The administrator can remove users via the administration program.

Termination information must be passed on if the users exist as proxied sessions on downstream RADIUS servers because these servers must free the resources previously allocated to the session(s), which have now been terminated.

The Proxy AutoStop feature is designed to handle such cases. In addition, if you use the LCI command to free resources from the central administrative server, the appropriate message are propagated so that the resources associated with the user in each of the downstream servers are automatically freed.

Tunnels

This section provides background information about tunnels and explains how to configure the Steel-Belted Radius server to support them.

Note: Steel-Belted Radius does not add tunnel functionality to your network. Steel-Belted Radius is able to support the authentication and accounting needs of any tunnels that you've already set up.

A *tunnel* is a uniquely secure type of remote connection. A tunnel passes data between a remote site and an enterprise site, providing an additional layer of encrypted protocol “wrapper” around the data. A tunnel offers authentication and encryption features that help secure the connection against network vandals and eavesdroppers. In addition, it can provide quality of service features such as guaranteed bandwidth.

All administration and configuration of the tunnel happens at the remote site. This is the side of the connection that requests remote access and opens the tunnel. An administrator at the remote site must configure the tunnel with various attributes: its destination IP address, what security protocols it supports, its password, and so on. These attributes are stored in a database to be retrieved when needed to set up a connection. It is useful to centralize the information by storing the tunnel attributes on a RADIUS server.

At connection time, the tunnel is established by a NAS device at the remote site. The NAS retrieves the tunnel configuration attributes from the RADIUS server and uses them to open the tunnel into the enterprise. After the tunnel is open, the user can be authenticated at the enterprise.

A RADIUS server is said to “support tunnels” if it has the ability to store and retrieve the configuration data that a NAS needs to open a tunnel. Steel-Belted Radius fully supports tunnels. It can:

- Determine from the attributes in the incoming Access-Request whether the connection request involves a tunnel, and if so, which tunnel.
- Store and retrieve tunnel configuration data.
- Track the number of tunnels currently in use, compare to a maximum number, and refuse the connection if the number is exceeded.

Tunnel Authentication Sequence

The tunnel authentication sequence begins when an Access-Request arrives at the Steel-Belted Radius server:

- 1 Steel-Belted Radius checks if the Access-Request contains a Called-Station-Id attribute. If so, Steel-Belted Radius searches its database for a Tunnel entry that contains the indicated telephone number in its Called-Station-Id list.

Note: If realms are in use, Steel-Belted Radius also searches for this number in its realm configuration files. If a match is found, the Access-Request is routed to the realm, and the quest for a tunnel is abandoned. For this reason, it is important to ensure that DNIS numbers are unique across all Tunnel entries and across all realm configuration files.

If a match between the Called-Station-Id and a Tunnel entry can be found, Steel-Belted Radius constructs an Access-Accept message using the Attributes list in the matching Tunnel entry. It then returns the Access-Accept to the client NAS.

- 2 Steel-Belted Radius next checks if the Access-Request contains a username in the form *User<Delimiter>TunnelName* or *TunnelName<Delimiter>User*.

<Delimiter> is a single character that must match the server's tunnel delimiter character. The order of the realm name relative to the user name must match the server's tunnel naming convention (prefix or suffix). Both of these values are determined per server (that is, all tunnels that use this server must follow the same conventions) by entering them in the Configuration dialog.

Steel-Belted Radius searches its database for a Tunnel entry whose Tunnel name matches the incoming TunnelName.

If a match can be found, Steel-Belted Radius constructs an Access-Accept message using the Attributes list in the matching Tunnel entry. It then returns the Access-Accept to the client NAS.

- 3 If Steel-Belted Radius was able to match the Access-Request with a Tunnel entry, the NAS uses the attributes returned in the Access-Accept message to open a tunnel into the enterprise site. Authentication of the User-Name is attempted, usually at the enterprise site. If user authentication succeeds, the connection is complete. Otherwise, the user's connection request is denied.

If no matching Tunnel entry was found in steps 1 or 2, Steel-Belted Radius concludes that a tunnel is not involved in making this connection. It then continues with its User-Name parsing sequence determine a destination for the authentication request.

See "Request Routing" on page 60.

The following is a wildcard example for IP Addresses:

```
NAS-IP-Address = 199.100.10.0
```

where `NAS-IP-Address` indicates any IP address on the 199.100.10.0 network.

Configuring Tunnel Support

To configure the Steel-Belted Radius server to support a tunnel, you must open the Tunnels dialog in the Administrator program and add a Tunnel entry.

See “Tunnels Dialog” on page 118.

A Tunnel entry allows you to specify a list of connection Attributes such as the tunnel password, the IP address of the NAS at the enterprise site, encryption conventions to use, and so on. You can also enter the maximum number of tunnels that can be open at one time. You’ll need to coordinate with the administrator at the enterprise site to get some of this information.

Called Station Id

DNIS (Dialed Number Information Services) refers to a capability that many NAS devices have to determine and use the telephone number that was dialed to make a connection request. The RADIUS standard supports DNIS by specifying the following attributes:

- `Calling-Station-Id` is the number from which the user originated the request.
- `Called-Station-Id` is the telephone number that was dialed to make the network connection.

When setting up a Tunnel entry for the Steel-Belted Radius database, you can enter a telephone number or list of numbers in the **Called Station Id** list box on the Tunnels dialog. This list box identifies `Called-Station-Id` attribute values that the server should expect to find in tunnel connection requests.

Dictionaries for Tunnel Support

The Tunnels dialog allows you to create the Attributes list by selecting attributes from a drop-down list. The available selections include attributes from all standard and vendor-specific RADIUS dictionaries installed on the Steel-Belted Radius server.

Whenever the server is able to accept a tunnel connection request, it consults the corresponding Tunnel entry for the list of Attributes to return in the Access-Accept packet. Steel-Belted Radius always returns any standard RADIUS attributes that appear in the Attributes list. It also returns any vendor-specific attributes that are appropriate for the Make/model of the NAS that requested the tunnel connection.

Vendor-specific attributes that appear in the Attributes list, but that do not apply to the requesting NAS, are ignored.

IP Address Assignment

Steel-Belted Radius can assign IP addresses in one of the following ways:

- **Static assignment.** Each time the user connects, the same specific address is assigned. For example, if the user `Kevin` has the `Framed-IP-Address` attribute set to `123.11.245.123`, then each time Kevin connects to the network, the IP address `123.11.245.123` is assigned.
- **Assignment from a specific address pool.** When the user connects, an address is assigned from a specific pool. For example, if user `Kevin` has `Framed-IP-Address` set to the `Sales` IP address pool, when Kevin connects to the network, the next available IP address from `Sales` is assigned.
- **Assignment from the RAS Client's IP address pool (or set of IP address pools).** When the user connects, an address is assigned from one of the pools associated with the RAS Client that makes the connection. For example, let's say that:
 - a RAS Client called `NAS1` uses IP address pool A;
 - a RAS Client called `NAS2` uses IP address pool B; *and*
 - a User entry called `Kevin` has a `Framed-IP-Address` attribute value of `pool` associated with RAS Client.

In this case, on connecting to the network, if user `Kevin` gets a port on `NAS1`, an IP address from pool A is assigned. On the next call, Kevin might connect to `NAS2`; in this case an address from pool B is assigned.

Alternatively, if a user has been assigned to a particular NAS-Specific IP Address Pool (and suffix), an IP address from that pool is assigned.

- **Assignment from DHCP server.** When the user connects, an address is assigned (leased) from a DHCP server for a user-configurable period of time. This period should be significant (for example, twenty-four hours).

See "Specifying IP Address Assignment in User/Profile Records" on page 122.

Hints

Steel-Belted Radius can treat the attribute `Framed-IP-Address` as a *Hint*. This means that if this attribute appears in the `Access-Request` and the user return list is configured to allocate `Framed-IP-Address` from a pool, the IP address in the `Access-Request` is returned instead of the newly-allocated IP address.

This functionality is defined in the `[Configuration]` section of `radius.ini`:

```
[Configuration]
FramedIPAddressHint = <yes/no>
```

When hints are enabled, Steel-Belted Radius uses a hint to determine the value of the `Framed-IP-Address` attribute in the access response. This means that `Framed-IP-Address` in the `Access-Request` is returned in the `Access-Accept`, regardless of the `Framed-IP-Address` value stored in the user's account.

The default value is `no`.

The following table details the effect of hints:

Account Configuration	Framed-IP-Address returned without hints	Framed-IP-Address returned with hints
No <code>Framed-IP-Address</code>	No value	<code>Framed-IP-Address</code> from <code>Access-Request</code>
Static Address	Static address	Static address
Address from Pool	Next address from pool	<code>Framed-IP-Address</code> from <code>Access-Request</code>

Note: By using hints, you can assign the same IP address to multiple active accounts.

Resource Management

This section explains how Steel-Belted Radius manages limited resources, such as network addresses, user or tunnel connections, and UDP ports.

Network Address Assignment

The Steel-Belted Radius address pooling feature allows you to set up one or more pools out of which unique network addresses are assigned dynamically as users require them. Each pool consists of a list of one or more ranges of IP addresses (an IP pool) or IPX network numbers (an IPX pool).

By using this feature, you can avoid allocating specific fixed addresses to individual users. You can make fewer addresses go farther, and you can consolidate address assignment across all your NAS devices.

How Address Assignment Works

Proper operation of address assignment from a pool depends crucially on both RADIUS authentication and RADIUS accounting transactions, as follows:

- 1 During the RADIUS authentication transaction, if the user's attribute settings specify address assignment from a pool, an address is allocated for that user from that pool.
- 2 The address is reserved for that user until a RADIUS accounting transaction indicates that the user has terminated the connection.

For this reason, the NAS device must be configured for RADIUS accounting, and the same Steel-Belted Radius server must be specified for both authentication and accounting. If your NAS is not configured for accounting (or does not support accounting), you cannot use the address pooling feature because addresses would be assigned but never released.

Setting Return-List Attributes

The `Framed-IP-Address` (or `Framed-IPX-Address`) `Return-List` attribute controls how the user's IP (or IPX) address is assigned. For each user known to the Steel-Belted Radius Administrator program, the `Framed-IP-Address` or `Framed-IPX-Address` attribute can be set.

Handling Address Leaks

Under optimal conditions, the system takes care of assigning and releasing addresses without any need to intervene. But in some circumstances, you can get *address leakage*; that is, an address remains reserved for a user after the user has terminated the connection.

Address leakage occurs when the address has been assigned during the authentication transaction, but the accounting transaction that would have released the address is never received by Steel-Belted Radius. This can occur for several reasons:

- The Steel-Belted Radius server might have been taken down for a period of time during which accounting transactions occurred.
- The NAS device might have crashed or been taken down before the user terminated. (In many cases, however, Steel-Belted Radius might be able to

prevent address leakage by recovering the addresses when the NAS starts up again.)

- The NAS might have sent the authentication and accounting transactions to a different RADIUS servers.
- Despite a successful authentication, the user's PPP negotiation with the NAS might have terminated unsuccessfully for a variety of reasons. In such a case, some NAS devices might not initiate a subsequent accounting transaction.
- Routing problems might have prevented the accounting transaction from reaching Steel-Belted Radius.

An address that has “leaked” remains out of circulation until you manually release it by displaying the Sessions list and deleting the corresponding session.

See “Deleting Entries from the Sessions List” on page 160.

Address Leakage Upon Stopping and Starting the Server

Steel-Belted Radius maintains all current address assignments in a persistent database on disk. If you shut down the server and then restart it, all the information about which address is assigned to which user is retained.

Note that if you leave Steel-Belted Radius turned off for a substantial period of time after addresses have already been assigned, you run the risk of address leakage as described above. When you start the server up again, be sure you review the Current Users dialog and delete any entries you know to be obsolete.

Overlapping Address Ranges

If you maintain multiple IP or IPX address pools, you can duplicate some of the addresses among the pools. The address tracking mechanism of Steel-Belted Radius, when it is enabled, ensures that, if an IP address appears in more than one pool, after it is assigned out of any pool, it remains unavailable through any of the pools until it is released.

You must disable this type of address tracking if the server is assigning IP addresses from disjoint networks. In that configuration, two numerically identical IP addresses would signal a conflict, even though they actually belong to two different networks.

Order of Address Assignment

IP or IPX addresses are assigned on a FIFO basis; that is, the address that was first released is the first to be reassigned. This ensures that addresses are out of use for as long as possible prior to reuse.

Concurrent Network Connections

The Steel-Belted Radius Administrator program allows you to limit the number of active connections, on a per-user, or per-tunnel basis.

Concurrent User Connections

You can set a maximum limit on the total number of concurrent connections that a user can have. Subsequently, when the user requests a new connection, Steel-Belted Radius compares the current number of connections to the maximum limit. If a new connection would exceed the limit, Steel-Belted Radius can either:

- Reject the additional connection; *or*
- Allow the connection, but log the event in the Radius event log (described in “Radius Log File” on page 144).

Note: When counting connections, Steel-Belted Radius does not distinguish between multi-link connections and new user authentication attempts.

For concurrent connection limits to work, it is essential that each NAS be configured for RADIUS accounting and that the same Steel-Belted Radius server be responsible for both authentication and accounting. These conventions give the server full access to the data it needs to accurately track connections.

The maximum number of concurrent connections can be set individually for any User entry of any User-type. The concurrent connection limit is set in the Users dialog by selecting the **Maximum Concurrent Users** checkbox and entering a number in the accompanying field.

See “Users Dialog” on page 91, especially “Concurrent Connection Limits” on page 99.

For individual users, a limit applies to the user; for groups, a limit applies to all members of the group. For example, if GroupA has a connection limit of 2, then users \\GroupA\userID1 and \\GroupA\userID2 (on an NT-based server) are each entitled to 2 concurrent connections.

Authentication methods that do not require User entries must provide alternate mechanisms for supporting concurrent connection limits. For example, if you are using external database authentication there is an alias mechanism you can use in the SQL or LDAP configuration file. Concurrent connection limits can be supported under proxy authentication only if the target server supports them.

Note: Concurrent user connections can be tracked across multiple Steel-Belted Radius servers by adding the Concurrency Server package.

Concurrent Tunnel Connections

The Steel-Belted Radius server uses its Sessions list to determine the number of active connections for each Tunnel. The Sessions list summarizes all of the RADIUS accounting data currently available to the server. Tunnel connections appear in the Sessions list using a special display convention that distinguishes them from user connections.

You can set a maximum limit on the total number of concurrent connections that can be open using a specific Tunnel. Subsequently, when a user requests a new connection via that Tunnel, Steel-Belted Radius compares the current number of connections to the maximum limit. If a new connection would exceed the limit, Steel-Belted Radius rejects the additional connection.

For concurrent connection limits to work, it is essential that each NAS that can open a tunnel be configured for RADIUS accounting and that the same Steel-Belted Radius server be specified for both authentication and accounting. This permits the server's Sessions list to be kept up to date and available to every NAS that needs to authenticate tunnel connections.

Note: Concurrent tunnel connections cannot be tracked across multiple Steel-Belted Radius servers without additional software extensions. Contact Funk Software for more information.

Attribute Value Pooling

The Attribute Value Pooling feature of Steel-Belted Radius allows you to define pools of attribute sets that are assigned when an Authorization Request is processed. The attribute sets are distributed according to specified weights and the values are returned with the Access-Accept.

This technique allows for a dynamic allocation of attribute values sets, so that attributes needed to configure changeable and complex situations do not have to be assigned in static profiles.

See “Attribute Value Pools (*.rr files)” on page 247 for details.

Phantom Records

The Steel-Belted Radius server allocates certain limited resources to its clients; these resources include IP addresses, IPX addresses, user connections, and tunnel connections.

Each time a client is allocated a resource, the Steel-Belted Radius server generates a *phantom* accounting record for its internal use. Phantom records are not written to the RADIUS accounting database, but they are displayed in the Sessions List

window, where they closely resemble accounting start records; the only difference is that the phantom records display N/A in the Session-ID column.

See “Sessions List” on page 158.

After the Steel-Belted Radius server receives the corresponding accounting start request packet from the client, the phantom record is no longer needed. Steel-Belted Radius discards the phantom and, in the Sessions List display, replaces N/A with the actual `Session-ID` number returned by the client device.

In some cases, a user can be allocated a resource and a phantom record can be created, but the Steel-Belted Radius server might never receive a corresponding start packet from the client. Since the resource is tied up from the moment it is allocated to the user, it is desirable to limit the amount of time spent waiting for the start packet to confirm the transaction.

The default time that the Steel-Belted Radius server waits is 180 seconds. You can modify this time by editing the `radius.ini` file.

See “`radius.ini` [Configuration] Section” on page 212.

Technical Bulletins

For information about special features that have been added to Steel-Belted Radius, see Appendix B, “Technical Bulletins.”

Administration

4

- Administrator Program
- Servers Dialog
- RAS Clients Dialog
- Users Dialog
- Profiles Dialog
- Proxy Dialog
- Tunnels Dialog
- IP Pools Dialog
- IPX Pools Dialog
- Access Dialog
- Configuration Dialog
- Import/Export Capabilities

Administrator Program

The Steel-Belted Radius Administrator (`radadnt.exe` under Windows) lets you control all aspects of Steel-Belted Radius. In minutes, you can set up new users, alter standard profiles, or configure new NAS devices from any computer on the network.

Note: Administrators making large-scale changes to the Steel-Belted Radius database might prefer to use the LDAP command line interface. See “LDAP Configuration Interface” on page 332.

Running the Administrator

To run the Steel-Belted Radius Administrator program:

- **Windows:** Double-click the **RADIUS Administrator** icon.
- **UNIX:** Open the `index.html` or `default.htm` file in your browser. This file is located in the `java` subdirectory under the `admin` directory that you defined when you installed Steel-Belted Radius on the UNIX machine, usually at path `/radadmin/java`.

The Administrator’s main window opens and displays the Servers dialog. Radio buttons on the left of the main window allow you to select the dialog you want to display. However, you must use the Servers dialog to connect to a specific Steel-Belted Radius server before you can use other dialogs.

If you are running **UNIX:** When you click **Connect**, you are prompted to enter an administrative account name and password. Do so and click **OK**.

Dialogs and menu items are available to you only if the account under which you’ve connected to the server has the right to access those items.

Help with the Administrator

To get help with the Steel-Belted Radius Administrator:

- Under **Windows:** Press **[F1]** or select **Help > Topics** from the Administrator menu.
- Under **UNIX:** Click the **Help** button at the bottom of any Administrator dialog.

To identify the current version of the Steel-Belted Radius Administrator:

- Under **Windows:** Select **Help > About** from the Administrator menu.
- Under **UNIX:** Click the **About** button just beside the **Help** button.

Exiting the Administrator

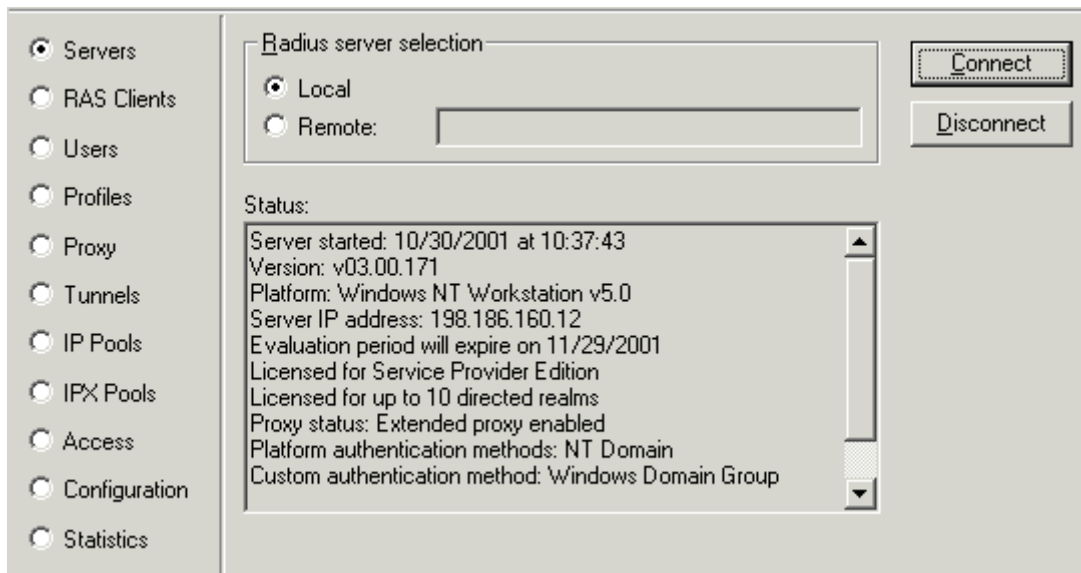
To close the Administrator program:

- **Windows:** Select **File > Exit**.
- **UNIX:** Select the Servers dialog and click **Disconnect**.

Closing the Administrator has no impact on the Steel-Belted Radius service or daemon.

Servers Dialog

The Servers dialog lets you select which Steel-Belted Radius server to administer within your network.



Servers dialog (Windows version)

After you connect to a server, the **Status** panel lists various features of the running server, such as version, platform on which it is running, IP address, available authentication methods, license information, and any initialization errors that might have occurred.

Steel-Belted Radius allows you the administrative rights that have been assigned to the account under which you connect to the server. These rights govern which server settings you can view or change.

UNIX

The **Radius Server Selection** panel lets you choose which server to administer. To use this panel, do one of the following:

- If you are running Steel-Belted Radius on your local host, click **Local** and click **Connect**.
- If you are running Steel-Belted Radius on a remote host, click **Remote**, enter the name of the remote host, and click **Connect**.

When you click **Connect**, the server prompts you to enter an administrative account name and password. This can be:

- The default Steel-Belted Radius administrative account (**admin**), whose password you can configure using the Access dialog.
- Any UNIX user or group account that you have enabled for Steel-Belted Radius administration using the **admin.ini** and **access.ini** configuration files.

Enter the account name and password in the dialog, then click **OK**.

Windows

The **Radius Server Selection** panel lets you choose which server to administer. To use this panel, either:

- If you are running Steel-Belted Radius on your local host, click **Local** and click **Connect**. The Administrator program verifies that the account under which you logged into the local machine has been enabled for Steel-Belted Radius administration using the Access dialog or the **admin.ini** and **access.ini** configuration files.
- If you are running Steel-Belted Radius on a remote host, click **Remote**, enter the name of the remote host, and click **Connect**. The Administrator program examines the remote machine for an NT user or group account that matches the username and password under which you logged into the local machine. If found, it verifies that this account has been enabled for Steel-Belted Radius administration using the Access dialog or the **admin.ini** and **access.ini** configuration files on the remote machine. If a match is found, the Administrator lets you connect to the remote machine under this account.

RAS Clients Dialog

The RAS Clients dialog lets you identify the devices that you want to define as clients of the Steel-Belted Radius server.

Servers

RAS Clients

Users

Profiles

Proxy

Tunnels

IP Pools

IPX Pools

Access

Configuration

Statistics

Client name: ASCEND MAX

IP address: 209.46.103.141

Make/model: - Standard Radius -

Use different shared secret for accounting

Assume down if no keepalive packets after (seconds):

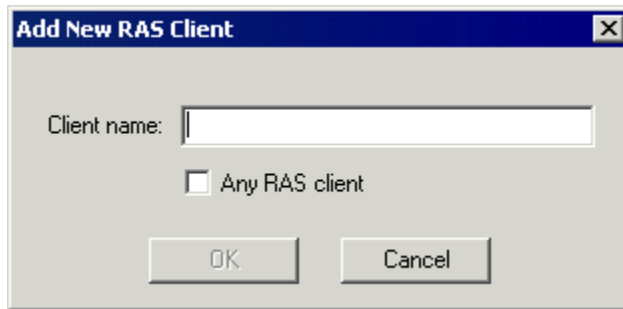
IP address pool: NEW ENGLAND

RAS Clients Dialog (Windows version)

Adding a New RAS Client

To add a new RAS Client:

- 1 Click the **Add** button. The Add New RAS Client dialog appears.



- 2 Enter the name of the RAS Client you'd like to add.

Although you can assign any name to a RAS Client entry, you should use the device's IP host name or hostname to avoid confusion.

- 3 Click **OK** to return to the RAS Clients dialog.

The name appears in the **Client name** field, with blank settings beneath.

- 4 Edit the settings, being sure to fill in all required fields.
- 5 Click **Save** to make your changes permanent.

Editing RAS Client Settings

After you edit the RAS Client settings, click **Save** to make your changes permanent. If you change your mind and want to cancel your edits, click **Reset**.

IP Address

Enter the RAS Client's IP address directly into the **IP Address** field. You can enter the DNS name of the device; the name you entered is resolved and the corresponding IP address is entered automatically into the **IP Address** field.

Make/model

The **Make/model** field offers a drop-down list from which you can select the make and model of the client device (Ascend MAX Family, Nortel CVX 1800, and so forth). Your **Make/model** selection tells Steel-Belted Radius the correct dictionary of RADIUS attributes to use when communicating with this client.

See “Dictionary Files” on page 239.

To select the Make/model:

- 1 For information about the various brands of NAS device supported by Steel-Belted Radius, click the **Vendor Info** button to display a detailed help file.
- 2 Click the **Make/model** drop-down box to bring up a list of the available NAS device makes and models.
- 3 Scroll through the list and select the item you want in the **Make/model** field. If you are not sure which make and model you are using or if your device is not in the list, select - **Standard Radius** -.

IP Address Pool

The **IP Address Pool** field specifies the pool from which Steel-Belted Radius selects IP addresses when authenticating an access request from this RAS Client. This field is optional and can be left blank.

To associate the RAS Client with an IP address pool:

- 1 Click the **IP Address Pool** drop-down box to bring up a list of previously configured IP address pools.
- 2 Scroll through the list and select the item you want in the **IP Address Pool** field.

Note: Only IP address pools that have been configured, by the IP Pool dialog and other means, appear on the list. See “IP Pools Dialog” on page 121.

Shared Secret

See also “RADIUS Shared Secret” on page 34.

To enter the shared secret for authentication:

- 1 Click **Edit authentication shared secret**. The Shared Secret dialog appears.
- 2 To enter a shared secret, simply type it into the dialog and click **Set**.
- 3 For privacy, asterisks are echoed as you type. But if no one is looking over your shoulder, you can check **Unmask shared secret** to see what you’re typing and make sure it is correct.
- 4 These steps complete configuration of the authentication shared secret on the server side. Be sure to enter the same authentication shared secret when you configure the NAS device.

To enter the shared secret for accounting:

- 1 If you want to use the same shared secret for authentication and accounting, ensure that the **Use different shared secret for accounting** box is unchecked before you click **Save** in the RAS Clients dialog.
- 2 Otherwise, enter a secret for accounting as follows: In the RAS Clients dialog, check the **Use different shared secret for accounting** box and click **Edit accounting shared secret**. Enter the accounting shared secret into the pop-up dialog and click **Set**.
- 3 These steps complete configuration of the accounting shared secret on the server side. Be sure to enter the same accounting shared secret when you configure the NAS device.
- 4 If you ever need to verify a shared secret on the Steel-Belted Radius server side, in the RAS Clients dialog click the appropriate **Edit** button, enter the shared secret, and click the **Validate** button. You'll be told whether the shared secret is what you think it is.

Assume Down

If you check the **Assume down if no keepalive packets** box, you can enter a value in the **after (seconds)** field. If the server does not receive any RADIUS packets from this client after the specified number of seconds, it assumes that the client device has gone down.

Steel-Belted Radius then gracefully closes any user or tunnel connections it has authenticated for this device. That is, Steel-Belted Radius releases any pooled IP or IPX addresses and adjusts the counts of concurrent user or tunnel connections appropriately.

Warning: Give thought to the value that you set for this field. If the after (seconds) value is set too small, valid user or tunnel connections can be lost. For example, during low usage periods, many NAS devices might send no RADIUS packets to the Steel-Belted Radius server; however, these devices are still “up.”

Adding the <ANY> RAS Client

A special RAS Client entry, called <ANY>, allows Steel-Belted Radius to accept requests from any NAS or Proxy RADIUS server, as long as the shared secret is correct.

To add an <ANY> entry:

- 1 Click **Add**. The Add New RAS Client dialog appears.

- 2 Check **Any RAS Client**, then click **OK**.
You now see <ANY> in the **Name** field, with blank settings beneath.
- 3 Update the Make/model and shared secret(s) for this item, then click **Save** to make your changes permanent.

Note that the **IP Address** field cannot be edited. <ANY> implies that the server accepts requests from any IP address, provided that the shared secret is correct.

Removing a RAS Client

To remove a RAS Client from the list:

- 1 Click the **Name drop-down** list and select the RAS Client you would like to remove.
- 2 In the RAS Clients dialog, click **Remove**.
- 3 You are prompted to confirm the operation. Click **Yes**.

Users Dialog

The Users dialog lets you configure RADIUS authentication details. Each User entry in the Steel-Belted Radius database identifies one method by which the server can authenticate a specific user. The **User name** field identifies the user; the **User type** field identifies the authentication method.

See “Authentication Methods” on page 36 and “Configuring Authentication Methods” on page 38.

There is more than one way to populate the User database for Steel-Belted Radius. You can use the Administrator program, Users dialog, as described in this topic. Alternatively, you can import data from other servers.

See “Import/Export Capabilities” on page 138.

Methods of Domain Authentication (Windows only)

The first-generation of domain authentication, which was specific to Windows NT technology, has been augmented for Steel-Belted Radius by a second-generation domain authentication plug-in. This plug-in is a superset of the functionality of the first-generation authentication method, supporting domain user and group authentication. It works on Windows 2000 and XP operating systems, which are not supported by the first-generation domain authentication method, and it supports

MS-CHAP and MS-CHAP-V2 from anywhere in the domain. Authenticating a user by way of a domain server is a particularly efficient way to manage account information by avoiding the duplication of work.

The second-generation domain authentication method (called “Windows Domain Authentication”) is automatically enabled when you install the server. You can choose between this and the first generation domain authentication method (called “NT Domain Authentication”). Funk Software recommends that you migrate to Windows Domain Authentication.

The first-generation domain authentication method types appear in the Users dialog as **NT Domain User** and **NT Domain Group** options, while the second-generation domain authentication method types appear in the Users dialog as **Windows Domain User** and **Windows Domain Group** options.

To use the Windows Domain Authentication plug-in, the RADIUS service must be run under the `LocalSystem` account on an Windows NT4, 2000, or XP computer that is part of a domain. Groups and users to be authenticated can reside in any domain within the forest, as well as in those domains outside the forest for which a trust relationship exists.

This plug-in authenticates Domain Users and Groups entered in the database by means of the **Domain** tab in the **Add New User** menu of the Configuration user interface (provided that they have the `Logon-locally` set correctly). These are the same Domain Users and Groups that are authenticated by the first-generation (NT) Domain methods. Both types of methods can be present and activated simultaneously; methods are called in the order of their appearance in the **Configuration** panel of the Configuration user interface.

The authentication process of this plug-in is very similar to that of the NT Domain Authentication methods. The domain name can be present in the User-Name attribute of the Access Request, which can be of the form `\\domain\user`, `domain\user`, or simply `user`. Additionally, the form `user@domain` can be used (although this form is not supported by NT-RRAS).

Prequalification Checklists

By default, when Steel-Belted Radius uses NT domain membership to authenticate a user, it processes attributes for the first group that the user matches. The attributes consist of checklist and reply-list attributes, and checklist processing is performed to determine the user’s authorization rights after authentication succeeds.

If an enterprise sets up separate NT domain groups for different access methods (for example, one domain group for users accessing the network through a VPN and another domain group for users accessing the network through a WLAN Access Point) and then assigns users to more than one domain group (so that the users get

different permissions based on what access method they use), Steel-Belted Radius can authenticate the user against the first group the user matches and process the wrong attributes for that user, causing checklist processing to fail and the user's access to be rejected.

Prequalification checklists allow a site to perform checklist processing before it authenticates a user, so that the attributes returned by every group a user belongs to can be evaluated (and the appropriate membership chosen) before authentication proceeds.

Example: CandyCorp sets up two groups (WLAN_USERS and VPN_USERS) in the CORP domain and creates access policies for each. Mary is a member of both groups; when she accesses the corporate network through a WLAN Access Point, her traffic should be tagged for a specific VLAN, and when she accesses the corporate network through a VPN, an Ascend-Data-Filter should be sent to the VPN gateway to restrict the internal hosts she can reach.

- Without prequalification checklist processing, Steel-Belted Radius responds to Mary's connection through an Access Point by using the first domain group membership it finds (which might be VPN_USERS), authenticating Mary and returning the attributes associated with that group, and then rejecting Mary because post-authentication checklist processing fails when the group used for authentication (VPN_USERS) didn't provide the appropriate access attributes.
- With prequalification checklist processing enabled, Steel-Belted Radius responds to Mary's connection through the Access Point by running checklist processing *before* it authenticates Mary: Steel-Belted Radius tests each group to which Mary belongs to see if authentication and authorization will ultimately be successful. If checklist processing for a domain group fails, that group is skipped and the next group is tried; if checklist processing for all groups fails, Mary's access request is denied. If checklist processing successfully matches Mary to a domain group, authentication proceeds, and Mary's traffic is processed according to corporate policies (that is, it is tagged with the VLAN identifier appropriate for her WLAN access).

The application of prequalification checklist processing is not limited to domain groups. Prequalification checklists can be used to direct a user request to an appropriate domain user entry based on the presence of attributes in the user's request. For example, if a user's name ("ADMIN") is specified in an Access-Request and both \\CORP\ADMIN and \\LAB\ADMIN are listed in the Steel-Belted Radius database with the same password, prequalification checklist processing could be used to select the appropriate domain user object for authentication and authorization.

Note: Prequalification checklist processing can be relatively expensive in terms of processing time. Each access request might entail multiple database operations, since Steel-Belted Radius must potentially review every domain

group to find one with attributes that match the user's checklist requirements.

Prequalification processing is enabled through the PrequalifyChecklist argument in the [NTDomain] section of the radius.ini file. For more information on the PrequalifyChecklist argument, see “radius.ini [NTDomain] Section (Windows only)” on page 218.

MS-CHAP Considerations

If the user is successfully authenticated, any appropriate encryption keys (obtained through either MS-CHAP or MS-CHAP-V2) are returned to Steel-Belted Radius and the user's profile is retrieved from the Steel-Belted Radius database. To enable encryption, the appropriate attributes (e.g., Mppe-Send-Key and Mppe-Recv-Key) must be included in the user's profile.

There is no longer a three-second authentication timeout per domain. There is now, therefore, no need to prepend the username with the Domain name to avoid timeout when dealing with a large number of domains.

The Windows Domain Authentication plug-in does not support EAP pre-fetch.

Configuration

As with other authentication plug-ins, winauth.dll is configured through a single .aut file. To operate, it must contain [Bootstrap] and [NTDomain] sections as follows:

```
[Bootstrap]
LibraryName=w2kauth.dll
Enable=1
InitializationString=W2K / NT4 authentication
```

The initialization string is ignored when registering the plug-in's two methods, which are identified by the strings W2K / NT4 Domain User and W2K / NT4 Domain Group.

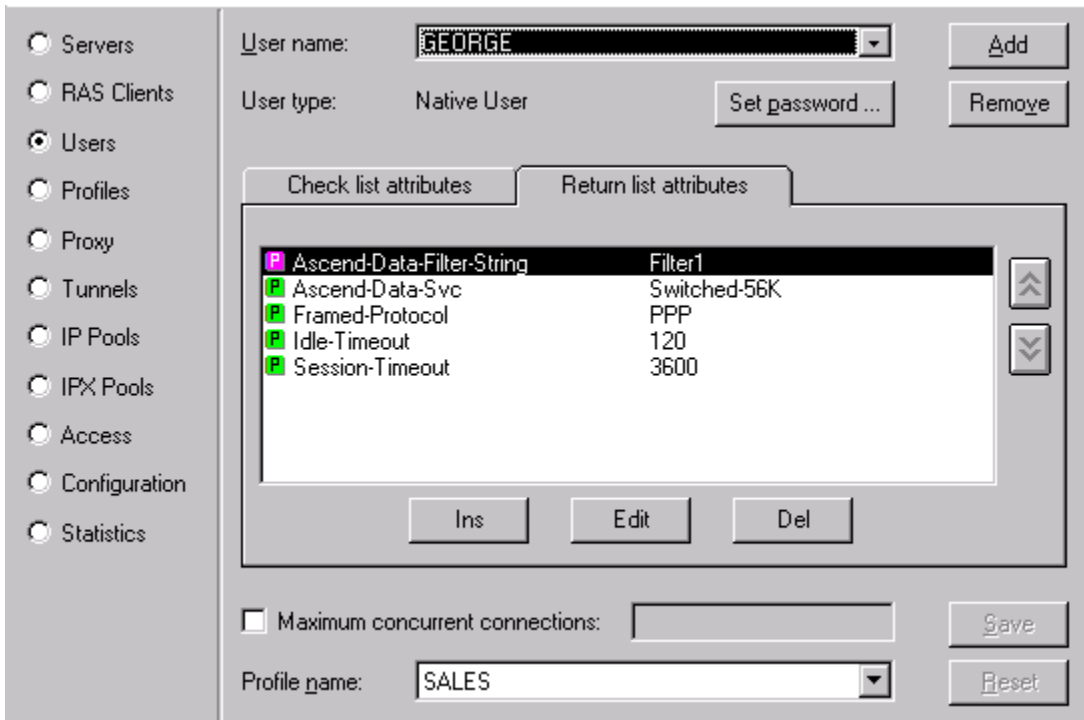
Handling of users with expired passwords is configured in the following section, similar to the configuration of NT Domain methods in radius.ini:

```
[NTDomain]
AllowExpiredPasswordsForUsers = yes
AllowExpiredPasswordsForGroups = yes
ProfileForExpiredUsers = Profile
ProfileForExpiredUsersInGroups = Profile
```

Note: MS-CHAP and MS-CHAP-V2 users with expired passwords are not accepted. They may be prompted to change password if their login application supports password changing.

Using the Users Dialog

The Users dialog allows you to manipulate the records of individual user accounts.



Users Dialog (Windows version)

To create a new User entry, click **Add** and follow the instructions in the next several sections.

Use the button to the right of the **User name** list box to select the User entry you want to view or edit. Users of all local types are listed together in alphabetical order. You can select a name from the list.

Windows

The name is displayed in the Users dialog, **User name** field. The user's other settings are displayed in the remaining fields of the Users dialog.

UNIX

The list of users in the **User name** field is not generated until you click the button. To avoid the time required to build the list, you can click the **Find** button if you know the specific name for which you want to search. When you click the **Find** button, a dialog prompts you to enter the name you want.

Pay attention to case when typing the name in the Find User dialog, or you might be unable to find the User entry. Native User entries in the Steel-Belted Radius database have all-uppercase names; the names are converted to all-uppercase letters when the Native User entry is created, and they remain all-uppercase for the life of the entry.

For example, a name entered as **realLife1** in the Add User dialog is stored as **REALLIFE1** in the Steel-Belted Radius database. Usernames stored in a database outside Steel-Belted Radius (UNIX, SecurID, TACACS+) retain their case as stored in that database.

After you've entered the name in the Find User dialog, click **OK**. The name is displayed in the Users dialog, **User name** field. The user's other settings are displayed in the remaining fields of the Users dialog.

Editing User Settings

This section describes fields that you can set for any User entry, regardless of User type. For more information, see “User Attribute Lists” on page 54 and “Profiles Dialog” on page 111.

Selecting a Profile

We strongly recommend that you make use of the powerful profile feature, rather than separately entering Check-List and Return-List attributes and values for each User entry.

To select a profile for a User entry:

- 1 Click the **Profile name** drop-down list.
- 2 Select the profile you'd like to use, or select **<no profile>**.

Adding New Attributes

To add Check-List or Return-List attributes to a User entry:

- 1 Click the **Check List Attributes** tab or **Return List Attributes** tab.
- 2 Click **Ins**.

- 3 The Add New Attribute dialog appears. You can add as many attributes as you want before closing this dialog. The dialog is positioned so that as you add attributes you can see them appear in the list.

Use the dialog as follows:

- Select an item from the list of **Available attributes**.
- You'll be able to tell whether you can add multiple values for this attribute by noting the state of the **Multi-Valued** indicator.
- Enter a value for the attribute in the space to the right of the attribute list. The method for entering a value varies depending on whether you are entering a string or number or are selecting from a list of options.
- (Check-List attributes only) If you want to set this value as the default value for the attribute in case the attribute is not included in the RADIUS request, check the **Default** box.
- (Return-List, single-valued attributes only) If you don't want to specify a particular value, but want to make sure that whatever value of the attribute appears in the RADIUS request is echoed to the client in the RADIUS response, check **Echo**.
- Click **Add** to add this attribute/value pair to the list.

- 4 Click **Close** to return to the User dialog.

Setting Attribute Values

To change the value of an attribute already in the Check-List or Return-List for a User entry:

- 1 Click the **Check List Attributes** tab or **Return List Attributes** tab.
- 2 Highlight the attribute whose value you'd like to change.
- 3 Click **Edit** or double-click the attribute. A Change dialog opens.

Depending on the attribute, you might be asked to enter a new value or to select a value from a list. For some attributes, Steel-Belted Radius retrieves the value from the server and you cannot enter a value in this dialog.

- 4 If prompted, enter or select the new value.
- 5 Click **OK**.

Removing Attribute/Value Pairs

To remove an attribute/value pair already in the Check-List or Return-List for a User entry:

- 1 Click the **Check List Attributes** tab or **Return List Attributes** tab.
- 2 Highlight the attribute/value pair you'd like to remove.
- 3 Click **Remove**. The value is removed from the list.



Reordering Attributes

Certain attributes are multi-valued and orderable; that is, the attribute/value pair can appear more than once in a RADIUS response, and the order in which the attribute/value pairs appear is important.

To reorder attributes in a User entry:

- 1 Click the **Check List Attributes** tab or **Return List Attributes** tab.
- 2 Highlight an attribute/value pair in the list.
- 3 Click one of the double-up and double-down arrows, as follows.

Note: These arrows are activated only if the attribute you've selected is both multi-valued and orderable; for example the standard RADIUS authentication attribute Reply-List.

Button	Action
	Moves the selected attribute/value up in the list. If the attribute is not orderable, or if this item is already the first value for this attribute, then the button is disabled.
	Moves the selected attribute/value down in the list. If the attribute is not orderable, or if this item is already the last value for this attribute, then the button is disabled.

Changing Attributes Inherited from a Profile

As noted previously, Check-List and Return-List attributes can be directly entered for any user, or they can be inherited from the profile selected as part of the user's settings.

Attributes inherited from a profile can be removed or modified locally; that is, any changes made to profile attributes for one user do not affect other users sharing the same profile.

Items that are inherited from a profile are marked with an icon:

Icon	Meaning
	Indicates the inherited attribute is unchanged
	Indicates the inherited attribute has been changed
	Indicates the inherited attribute has been removed

To change an attribute inherited from a profile, click **Edit** and proceed as you would normally. The modified attribute is marked with .

To remove an attribute inherited from a profile, click **Remove**. The attribute does not disappear from the display, but is shown grayed and struck-through, and is marked with .

To restore an attribute that you have changed or removed to its original value as specified in the profile, click **Remove**. The attribute is reset to its unmodified state, and is marked with .

Note: If you accidentally delete a profile being used as part of a User setting, you see an error icon displayed on all lines dependent upon this profile. If this occurs, delete the items in error and re-create the profile.

Concurrent Connection Limits

See “Concurrent Network Connections” on page 79.

A maximum number of open connections can be set for each User entry by checking the **Maximum Concurrent Users** box and entering a number in the accompanying field. When the user requests access, this user (User name) can be authenticated using this method (User type) only if fewer than this number of connections are currently open for this user.

Allowed Access Hours

The user’s allowed access hours can be specified by adding the `Funk-Allowed-Access-Hours` attribute to the user’s Check-List.

See “Allowed Access Hours” on page 43.

`Funk-Allowed-Access-Hours` is a variable-length string that identifies time periods in a 7-day week of 24-hour days. This string consists of one or more day specifiers (each of which can list one or more days and/or ranges of days) followed by one or more ranges of 24-hour times, in minutes. For example:

```
Funk-Allowed-Access-Hours M-W 0100-1400 2300-2400 Tu,Th  
M-Tu, F 0530-1500, Sa-Su 0000-2400.
```

The syntax rules for `Funk-Allowed-Access-Hours` are as follows:

- Time ranges can be inclusive (1000-1100 allows access only between 10 a.m. and 11 p.m.) or exclusive (1100-1000 allows access any time except between 10 a.m. and 11 a.m.).
- Days, specifiers, and ranges of days and times can be separated by commas or spaces; ranges of days or times are indicated by hyphens (m-w or 0239-1459).
- Days can be specified by the minimum number of case-insensitive letters necessary to distinguish them (Su, M, Tu, W, Th, F, Sa) and can wrap around the end of the week (Sa-Su).
- At least one time period is required for each day; that is, each day, list, or range of days must be followed by one or more ranges of times.
- Times are specified using four digits, with leading zeroes where needed (0001 for 12:01 a.m., 0630 for 6:30 a.m., and so forth).

When assigned to a user's Check-List, our sample `Funk-Allowed-Access-Hours` value (above) allows the user access during the following time periods:

- 1 a.m. to 2 p.m. and 11:00 p.m. to midnight, Monday and Wednesday
- 5:30 a.m. to 3:00 p.m. Monday, Tuesday, Thursday, and Friday
- Any time Saturday or Sunday
- The total access times Monday are 1 a.m. to 3:00 p.m. and 11:00 p.m. to midnight

If the user attempts access on Sunday at 11:30 p.m., access would be allowed and a `Session-Timeout` attribute specifying a value of 1800 seconds would be returned — 30 minutes (until midnight Sunday when the access period ends) times 60 seconds per minute. However, if the user's Return-List includes a `Session-Timeout` with a value less than 1800, this lesser value would be returned.

Adding a Native User

Native User entries require you to enter the user's name and password into the Steel-Belted Radius database. For all other types of User entry, the server relies on another database to confirm the user's password.

Under **Windows** only: You must define a Native User entry for every user who requires remote access to your network. For example, you can accommodate UNIX- or Macintosh-based users by adding them as Native Users.

To add a new Native User:

- 1 Click the **Add** button on the right. The Add New User dialog appears.
- 2 Select the **Native** tab.
- 3 Enter the User name into the field and click **OK**. The Add New Users dialog closes.

Back in the Users dialog, the **User name** field displays the name you've just entered. The **User type** field displays the authentication method Native User.

- 4 Click **Set Password**. The Enter User Password dialog appears.



Enter User Password dialog (Windows version)

Use the dialog as follows:

- If you'd like the actual password to be echoed as you type (rather than asterisks) check **Unmask password**. Enter the password for the new user.

Note: *The password is case-sensitive.*

- If you'd like to enable both PAP and CHAP authentication, check **Allow PAP or CHAP**.
- If you want your password to be stored using *strong encryption* in the Steel-Belted Radius database, check **Allow PAP only (encrypt password in database)**. This option allows the user to authenticate only via PAP. However, the server database is totally secure even if your server is compromised.
- To verify a password you've already entered, type the password in the dialog and click **Validate**. You'll be told whether the password is what you think it is.

- When you have completed any changes, click **OK**.
- 5 Back in the Users dialog, edit settings for the new entry.
 - 6 To make your changes permanent, click **Save**.
 - 7 To edit (or add) another User entry, select (or enter) a new User name and User type.

Adding a Domain User or Domain Group (Windows only)

Most medium to large Microsoft networking installations are organized into one or more Domains for security purposes. If your network is so organized, domain authentication should be preferred over host authentication for RADIUS purposes. If you have enabled the first-generation domain authentication method, you get the options **NT Domain User** and **NT Domain Group** in the Users dialog. If you have enabled the second-generation domain authentication method, you get the options **Windows Domain User** and **Windows Domain Group** in the Users dialog.

See “Methods of Domain Authentication (Windows only)” on page 91.

To use Domain authentication, the Steel-Belted Radius service must run on a Windows computer that is part of a Domain. It doesn't matter whether that machine is a workstation or server, nor does it matter whether the machine is a Domain Controller.

It is possible to authenticate against Domains other than the one in which the Steel-Belted Radius service is running, provided that the other Domain is trusted by the Domain of the RADIUS service. The trust relationship may not be mutual; the other Domain does not have to trust the RADIUS Domain .

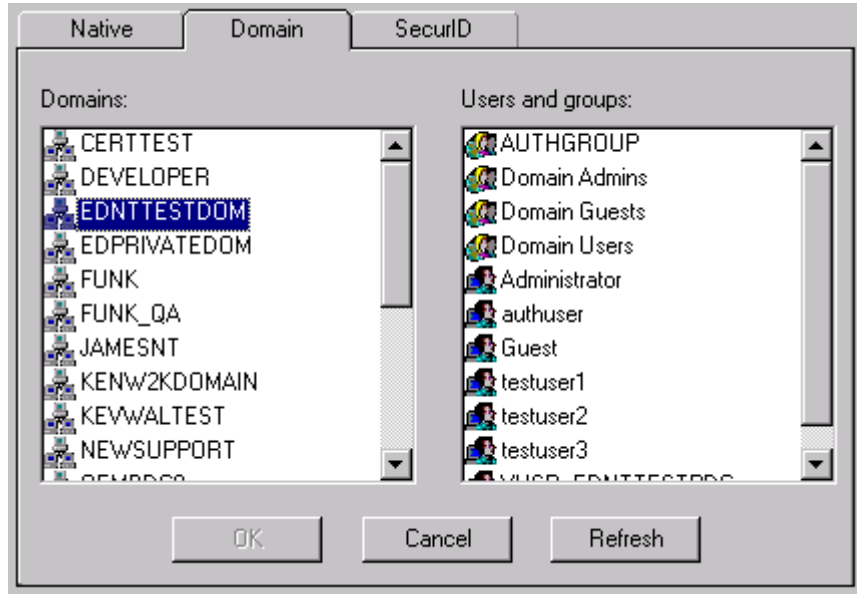
Example: Suppose there are three domains: A, B, and C, and Steel-Belted Radius is running in A. A trusts B and C trusts A. You'll be able to use Domains A and B for authentication, but not C. That is because A does not trust C.

You can add a Domain User entry to provide for the authentication of a specific user defined within a specific Domain under Microsoft networking. For more flexibility, you can add a Domain Group, to provide for the authentication of all users that belong to a specific group defined within a specific Domain.

To add a new Domain User or Domain Group:

- 1 Click the **Add** button on the right.
The Add New User dialog appears.

- 2 Select the **Domain** tab.



Add New Users Dialog, Domain Tab (Windows version)

- 3 Select a Domain name from the list on the left.
- 4 Now select a user or group from the list on the right.
- 5 Click **OK**.

Back in the Users dialog, the **User name** field now displays the user or group that you've selected, and the **User type** field displays the authentication method (Domain User or Domain Group).

- 6 Edit the settings for the new entry.
- 7 To make your changes permanent, click **Save**.
- 8 To edit (or add) another User entry, select (or enter) a new User name and User type.

Expired Domain Passwords

The Domain authentication methods allow users to be authenticated against Domain security using an expired Domain password.

This feature was developed to handle security policies that force Domain passwords to be changed automatically after a certain number of days. Typically, after the password expires, at the next attempt to log in, the Domain recognizes the password

supplied by the user as expired. The Domain then returns a special status code to its client application indicating these conditions. Typically, the user is then prompted to change his or her Domain password, but the client application (e.g., Microsoft Remote Access Client) must support the ability to change passwords.

Note: This password changing feature is supported only by Windows Domain Authentication (not NT Domain Authentication).

When Steel-Belted Radius passes a username/password pair through to a Domain for authentication, the Domain can indicate to Steel-Belted Radius that the password is expired. If so, Steel-Belted Radius's default response is to issue an Access-Reject. You can configure it to respond instead with an Access-Accept.

NT Domain Authentication Configuration

You can specify how the first-generation domain authentication method responds to an expired Domain password by editing the [NTDomain] section of the `radius.ini` configuration file. You can choose separate responses for NT Domain User and NT Domain Group authentication methods. Steel-Belted Radius takes the actions that you define in the [NTDomain] section whenever it receives either of the following status codes from an NT Domain:

- Expired password
- User must change password at next logon

Access-Reject is the least desirable action in either case, because it can prevent legitimate users from gaining access to the network and using billable services. However, in the case of an Access-Accept, Steel-Belted Radius must decide which Return-List attributes to provide. The remainder of this topic describes how these decisions are made.

Note: This section does not apply to Windows domain authentication, which uses a second-generation domain authentication method.

Let's say that the Domain User authentication method is being tried and Steel-Belted Radius finds that the incoming username matches a Domain User entry in its database. Steel-Belted Radius passes the username/password pair through to the Domain for authentication. If the Domain returns the `expired password` status code, Steel-Belted Radius has two options for specifying which Return-List attributes to include in an Access-Accept response:

Note: The first of the two options is strongly recommended.

- Steel-Belted Radius uses the Return-List from the corresponding Domain User entry in the Steel-Belted Radius database.

The basis for this decision is that, even though the password is out of date, it is still the correct password for this user in this Domain. Therefore, the user

should be allowed access to the network with all of his or her usual privileges (as indicated by the attributes in the Domain User's regular Return-List).

- Steel-Belted Radius uses a special-purpose Return-List. Steel-Belted Radius allows you to set up one Return-List that applies to all users who are accepted by the server under Domain User `expired password` conditions.

The basis for this decision is that any exception to normal behavior is a potential security risk. Therefore, the user should be allowed access to the network with a minimal set of privileges (as indicated by the attributes in the special-purpose Return-List).

Let's say that the Domain Group authentication method is being tried and there is at least one Domain Group entry in the Steel-Belted Radius database. Steel-Belted Radius passes the username/password pair through to the Domain for authentication.

If the Domain is able to authenticate the username/password pair, it gives Steel-Belted Radius a list of the groups of which the user is a member. If one of these groups matches a valid Domain Group entry, authentication proceeds as usual and the `expired password` feature is not needed.

If the Domain returns the `expired password` status code, it does not return a list of groups. In this case, Steel-Belted Radius must decide what to do. Potentially, the user is a member of one of the Domain Groups defined in the Steel-Belted Radius database. However, there is no way for Steel-Belted Radius to know for sure. Steel-Belted Radius has two options for specifying which Return-List attributes to include in an Access-Accept response:

Note: The first of the following two options is strongly recommended.

- Steel-Belted Radius uses a special-purpose Return-List. Steel-Belted Radius allows you to set up one Return-List that applies to all users who are accepted by the server under Domain Group `expired password` conditions.
- Steel-Belted Radius assumes that the user is a member of the first Domain Group (alphabetically) named in its database. It issues an Access-Accept using the Return-List from this Domain Group entry.

Note: After you determine the expired Domain password policies for your Steel-Belted Radius server, you must edit the `radius.ini` file on the server to implement them. For more information, see "radius.ini [NTDomain] Section (Windows only)" on page 218.

Adding a Host User or Host Group (Windows only)

Host authentication is provided to accommodate networks that aren't organized into Domains, or networks that mix Domain and Workgroup security. There are some

important restrictions and limitations to the use of Host authentication that you must be aware of if you intend to use it:

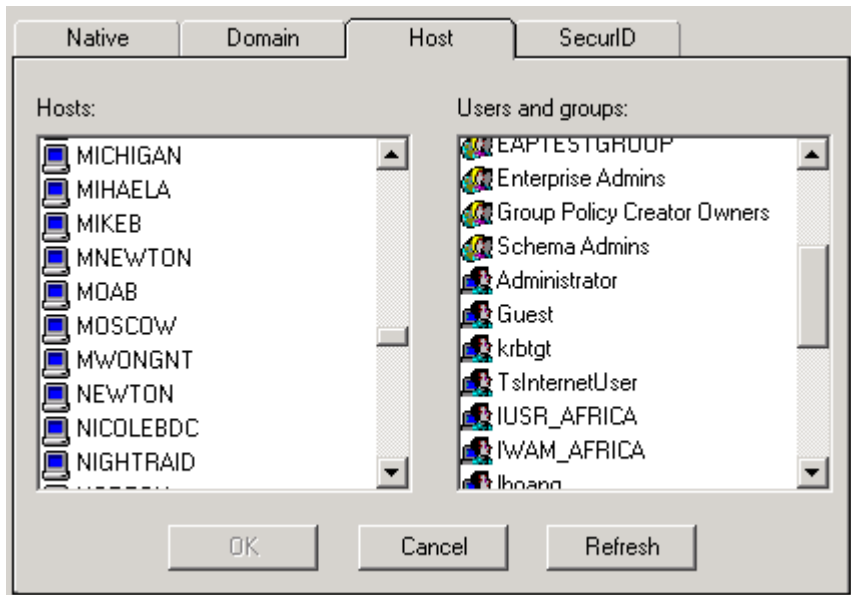
- The NT computer on which you install Steel-Belted Radius must not be a Domain Controller. Due to a limitation in Microsoft networking, peer-to-peer authentication does not work from a Domain Controller.
- The NT computer on which you install Steel-Belted Radius must not have peer-to-peer connections (for example, via a drive mapping) with other NT machines that might be used for Host authentication.

Microsoft networking permits only one peer-to-peer connection between two machines at a time. Steel-Belted Radius performs Host authentication by attempting to log the user into the specified Host; if the server is already logged into that Host (for any reason), the new login attempt fails due to the one-connection restriction.

You can add a Host User entry to provide for the authentication of a specific user defined on a specific Windows NT/2000 machine. For more flexibility, you can add a Host Group, to provide for the authentication of all users that belong to a specific group defined on a specific Windows NT/2000 machine.

To add a new Host User or Host Group:

- 1 Click the **Add** button on the right. The Add New User dialog appears.
- 2 Select the **Host** tab.



Add New Users Dialog, Host Tab (Windows version)

- 3 First select the name of an NT Host from the list on the left. Now select a Host user or group from the list on the right. Click **OK**.
Back in the Users dialog, the **User name** field displays the user or group that you've selected, and the **User type** field displays the authentication method (Host User or Host Group).
- 4 Edit the settings for the new entry.
- 5 To make your changes permanent, click **Save**.
- 6 To edit (or add) another User entry, select (or enter) a new User name and User type.

Adding a UNIX User or Group (UNIX only)

You can add a UNIX User entry to provide for the authentication of a specific user defined on the UNIX server. For more flexibility, you can add a UNIX Group, to provide for the authentication of all users that belong to a specific group defined on the server.

To add a new UNIX User or UNIX Group:

- 1 In the Users dialog, click the **Add** button on the right. The Add New User dialog appears.
- 2 Select the **UNIX** tab.
- 3 Select a user or group from the list. Click **OK**.
Back in the Users dialog, the **User name** field displays the user or group name that you've selected, and the **User type** field displays the authentication method (UNIX User or UNIX Group).
- 4 Edit the settings for the new entry.
- 5 To make your changes permanent, click **Save**.
- 6 To edit (or add) another User entry, select (or enter) a new User name and User type.

Adding a SecurID User

If you have an ACE/Server from Security Dynamics, Inc., Steel-Belted Radius can work with it to provide SecurID authentication for your users. First, you must set up communication between the two servers using the instructions in "Configuring SecurID Authentication" on page 24. Then, you can add SecurID users to the Steel-Belted Radius database using the instructions below.

Steel-Belted Radius attempts SecurID authentication only on usernames that match a SecurID entry in its User database. There are four types of SecurID entry, each providing a different matching rule:

- You can enter the name of a specific user.

For example, you might create a SecurID user entry for the specific user `George`. This tells Steel-Belted Radius that when an authentication request is received for username `George`, SecurID can be used as an authentication method and, if successful, the attributes of this user entry apply.

- You can enter a prefix.

For example, you might create a SecurID user entry for the prefix `sales$`. This tells Steel-Belted Radius that when an authentication request is received for a username such as `sales$Harry` or `sales$Cynthia`, SecurID can be used as an authentication method and, if successful, the attributes of this user entry apply.

Note: Only the part of the username after the prefix (Harry or Cynthia in the example above) is sent to the ACE/Server.

The advantage of using a prefix is that you don't have to create a separate user entry for each SecurID user; instead, you can group multiple SecurID users into a single user entry. Plus, if you want to use different settings for different groups, you can do that as well.

Note: The user must be sure to type in the prefix as part of the username he or she enters when dialing in.

- You can enter a suffix.

A suffix works just like a prefix, but appears at the end of the username; for example, if the suffix were `!sales`, you might have usernames such as `Harry!sales` or `Cynthia!sales`.

- You can create an entry for `Any` user.

This creates a single user entry named `<ANY>` that matches any username to be authenticated. SecurID can be used as an authentication method for any username, and, if successful, the attributes of the `<ANY>` entry apply.

The `<ANY>` entry makes sense if a single set of attributes apply to all your SecurID users and if you want to make SecurID either the only authentication method used or the authentication method of last resort if other authentication methods fail.

To add a new SecurID User entry:

- 1 Click the **Add** button on the right. The Add New User dialog appears.

- 2 Select the **SecurID** tab
- 3 Select the user type; either **Specific user**, **Prefix**, **Suffix**, or **Any user**.
- 4 Enter the specific user name, **<ANY>**, a prefix, or a suffix. Click **OK**.

Back in the Users dialog, the **User name** field displays the SecurID naming convention that you've selected. The **User type** field displays the authentication method (SecurID User, SecurID Prefix, or SecurID Suffix).

- 5 Edit the settings for the new entry.
- 6 To make your changes permanent, click **Save**.
- 7 To edit (or add) another User entry, select (or enter) a new User name and User type.

Each new suffix or prefix entry that you add appears in the Users dialog with the username represented by the string `USERNAME`, for example `!USERNAME` or `sales$USERNAME`.

Adding a TACACS+ User

If you have a TACACS+ Server, Steel-Belted Radius can work with it to authenticate your users. First, you must set up communication between the two servers using the instructions in “Configuring TACACS+ Authentication” on page 26. Then, you can add TACACS+ users to the Steel-Belted Radius database using the instructions below.

Steel-Belted Radius attempt TACACS+ authentication only on usernames that match a TACACS+ entry in its User database. There are four types of TACACS+ entry, each providing a different matching rule:

- You can enter the name of a specific user.

For example, you might create a TACACS+ user entry for the specific user `George`. This tells Steel-Belted Radius that when an authentication request is received for username `George`, TACACS+ can be used as an authentication method and, if successful, the attributes of this user entry apply.

- You can enter a prefix.

For example, you might create a TACACS+ user entry for the prefix `sales$`. This tells Steel-Belted Radius that when an authentication request is received for a username such as `sales$Harry` or `sales$Cynthia`, TACACS+ can be used as an authentication method and, if successful, the attributes of this user entry apply.

Note: Only the part of the username after the prefix (`Harry` or `Cynthia` in the example above) is sent to the TACACS+ server.

The advantage of using a prefix is that you don't have to create a separate user entry for each TACACS+ user; instead, you can group multiple TACACS+ users into a single user entry. Plus, if you want to use different settings for different groups, you can do that as well.

Note: The user must be sure to type in the prefix as part of the username he or she enters when dialing in.

- You can enter a suffix.

A suffix works just like a prefix, but appears at the end of the username; for example, if the suffix were `!sales`, you might have usernames such as `Harry!sales` or `Cynthia!sales`.

- You can create an entry for `Any` user.

This creates a single user entry named `<ANY>` that matches any username to be authenticated. TACACS+ can be used as an authentication method for any username, and, if successful, the attributes of the `<ANY>` entry apply.

The `<ANY>` entry makes sense if a single set of attributes apply to all your TACACS+ users, and if you'd like to make TACACS+ either the only authentication method used or the authentication method of last resort if other authentication methods fail.

To add a new TACACS+ User entry:

- 1 Click the **Add** button on the right. The Add New User dialog appears.
- 2 Select the **TACACS+** tab.
- 3 Select the user type; either **Specific user**, **Prefix**, **Suffix**, or **Any user**.
- 4 Enter the user name, `<ANY>`, a prefix, or a suffix. Click **OK**.

Back in the Users dialog, the **User name** field displays the TACACS+ naming convention that you've selected. The **User type** field displays the authentication method (TACACS+ User, TACACS+ Prefix, or TACACS+ Suffix).

- 5 Edit the settings for the new entry.
- 6 To make your changes permanent, click **Save**.
- 7 To edit (or add) another User entry, select (or enter) a new User name and User type.

Each new suffix or prefix entry that you add appears in the Users dialog with the username represented by the string `USERNAME`, for example `!USERNAME` or `sales$USERNAME`.

Removing a User Entry

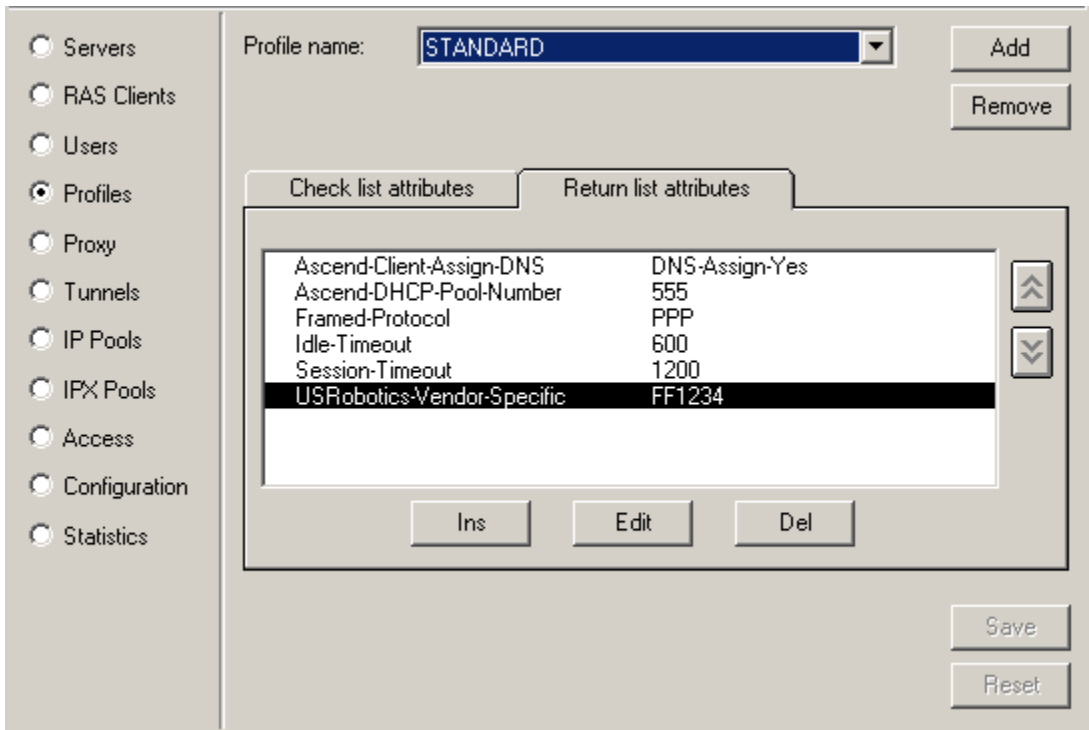
To remove a User entry:

- 1 Open the Users dialog.
- 2 Click the **User name** drop-down list and select the user you would like to remove. (Under **UNIX**: Click **OK** to proceed.)
- 3 In the Users dialog, click **Remove**. you are prompted to confirm the deletion.
- 4 Click **Yes**. The user is removed from the list.

Profiles Dialog

The Profiles dialog lets you define standard sets of Check-List and Return-List attributes. Any of these profiles can then be assigned to a User entry.

See also “User Attribute Lists” on page 54.



Profiles Dialog (Windows version)

Adding a Profile

To add a new profile:

- 1 Click **Add**. The Add New Profile dialog appears.
- 2 Enter a name for the new profile, and click **OK** to return to the Profiles dialog. The name you entered appears in the Profiles dialog **Name** field, with an empty attribute list below.
- 3 Add Check-List and Return-List attributes for the new entry.
- 4 Click **Save** to make your changes permanent.

Editing Profiles

The settings for each profile entry include Check-List and Return-List attributes. You can add, modify, and remove attributes in the Profile dialog just as you would in the Users dialog.

See “Editing User Settings” on page 96.

Removing a Profile

***Important:** Be sure you don't remove a profile that is currently included in the settings of a User. Steel-Belted Radius warns you if you try to delete a profile being used by one or more User entries. If you delete the profile anyway, the attributes defined in the profile disappear from that User's settings; when you next display the User's settings, an error message asks you to edit and resave those settings.*

To remove a profile from the list:

- 1 Click the **Name** drop-down list and select the profile you would like to remove. (Under **UNIX**: Click **OK** to proceed.)
- 2 In the Profiles dialog, click **Remove**. you are prompted to confirm the deletion.
- 3 Click **Yes**. The profile is removed from the list.

Proxy Dialog

The Proxy dialog lets you configure Steel-Belted Radius to forward RADIUS packets to another RADIUS server.

See also “Proxy RADIUS” on page 65.

The screenshot shows the 'Proxy' configuration dialog in a Windows application. On the left is a sidebar with radio buttons for 'Servers', 'RAS Clients', 'Users', 'Profiles', 'Proxy' (selected), 'Tunnels', 'IP Pools', 'IPX Pools', 'Access', 'Configuration', and 'Statistics'. The main area contains the following fields and controls:

- 'Forward to:' dropdown menu with 'DUALRAD' selected and an 'Add' button.
- 'IP address:' text box with '14.23.107.8' and a 'Remove' button.
- Two checkboxes: 'non-default authentication port:' and 'non-default accounting port:', each with an empty text box.
- 'Edit authentication shared secret ...' button.
- Checkbox: 'Use different shared secret for accounting'.
- 'Edit accounting shared secret ...' button.
- 'Retry policy' section with 'Number of retries:' (text box with '3') and 'Milliseconds between retries:' (text box with '5000').
- 'Proxy accounting' section with radio buttons: 'Forward', 'Record locally', and 'Forward and record locally' (selected).
- 'Authentication Method Status' section with checkbox: 'Include in authentication list'.
- 'Save' and 'Reset' buttons at the bottom right.

Proxy Dialog (Windows version)

Adding a New Target

This section explains how to set up proxy forwarding from the Steel-Belted Radius server (the proxy) to another RADIUS server (the target).

To add a new target server:

- 1 In the Proxy dialog, click **Add**. The Add New Target Server dialog appears.
- 2 Enter the name of the target server you'd like to add.

You can label a Proxy entry with any name you like. Steel-Belted Radius uses the Proxy entry's **IP Address** field to route the RADIUS packets correctly, so the actual node name of the target server is not important. The target name must not duplicate any other target name, realm name, or tunnel name in your Steel-Belted Radius configuration.

- 3 Click **OK**.
The name you entered appears in the **Forward to** field in the Proxy dialog, with blank settings beneath.
- 4 Edit the settings for the new target server entry. Be sure you fill in all required fields.
- 5 Click **Save** to make your changes permanent.
- 6 Ask the administrator at the target site to log into the target server's RADIUS configuration program and add Steel-Belted Radius as a RADIUS client of the target server. You'll need to provide this administrator with the IP address of the Steel-Belted Radius server.

Note: Make sure that the same UDP port and shared secret are entered on both proxy and target sides.

Editing Proxy Settings

The settings for each target server include the target server's IP address and a secret key that is shared between this proxy server and the target server.

After you edit the settings, click **Save** to make your changes permanent. If you change your mind and want to cancel your edits, click **Reset**.

IP Address

You can enter the IP address directly into the **IP Address** field. Alternatively, if you enter the DNS name of the target server, the name you entered is resolved and the IP address is entered automatically into the **IP Address** field.

UDP Ports

See "RADIUS Ports" on page 35.

You can enter the UDP port on which the target server receives RADIUS authentication traffic. If you do not already have this information, you'll need to acquire it from the administrator at the target site. To enter a port number, check the **non-default authentication port** box and enter a value in the accompanying field. If you do nothing, Steel-Belted Radius uses port 1645.

Similarly, you can enter a **non-default accounting port**. The default is 1646.

Shared Secret

See "RADIUS Shared Secret" on page 34.

To enter the shared secret for authentication:

- 1 Click **Edit authentication shared secret**. The Shared Secret dialog appears.
- 2 To enter a shared secret, simply type it into the dialog and click **Set**.
- 3 For privacy, asterisks are echoed as you type. But if no one is looking over your shoulder, you can check **Unmask shared secret** to see what you're typing and make sure it is correct.
- 4 These steps complete configuration of the authentication shared secret on the Steel-Belted Radius side. Be sure to enter the same authentication shared secret when you configure the target server.

To enter the shared secret for accounting:

- 1 If you want to use the same shared secret for authentication and accounting, ensure that the **Use different shared secret for accounting** box is unchecked before you click **Save** in the Proxy dialog.
- 2 Otherwise, enter a secret for accounting as follows: In the Proxy dialog, check the **Use different shared secret for accounting** box and click **Edit accounting shared secret**. Enter the accounting shared secret into the pop-up dialog and click **Set**.
- 3 These steps complete configuration of the accounting shared secret on the Steel-Belted Radius side. Be sure to enter the same accounting shared secret when you configure the target server.

If you ever need to verify a shared secret on the Steel-Belted Radius side, in the Proxy dialog click the appropriate **Edit** button, enter the shared secret, and click the **Validate** button. You'll be told whether the shared secret is what you think it is.

Retry Policy

When Steel-Belted Radius acts as a proxy, it needs to emulate the typical characteristics of a NAS device. This includes the ability to retransmit a request if it doesn't get a response within some interval of time.

There are two values that can be set:

- **Number of Retries**. This sets the number of times a request is retransmitted if an acknowledgment from the target is not received; if the number of retries is exhausted, then the original request is rejected.
- **Milliseconds Between Retries**. This sets the time interval between each retry in milliseconds (thousandths of a second). For example, a value of 2000 indicates that retries should occur every 2 seconds.

Proxy Accounting

The Proxy Accounting setting lets you control how accounting transactions are handled for authentication requests that are forwarded. There are three options:

- **Forward.** Forward the accounting transaction to the same target server that the authentication transaction was forwarded to.
- **Record locally.** Do not forward the accounting transaction. Log the accounting transaction locally even though the authentication request was forwarded.
- **Forward and record locally.** Do both. Forward the accounting transaction and log the accounting transaction locally.

Proxy Authentication

In the Proxy dialog, the Authentication Method Status panel lets you set up a target server as an authentication method. After you check the **Include in authentication** list box, the target that you've defined using the Proxy dialog appears in the Configuration dialog's **Authentication Methods** list as `proxy: name`, where *name* is the value you entered in the **Forward to** field.

This option is useful if you already have user records defined on an older RADIUS server and you want to provide a seamless migration to Steel-Belted Radius. You can set up the older server as a Proxy RADIUS target and check the **Include in authentication** list box. RADIUS requests that arrive addressed to this target are handled by Steel-Belted Radius automatically, without requiring end users to change their addressing conventions.

*Note: If the target that you're configuring is a member of a Proxy RADIUS realm, you should ensure that the Proxy dialog **Include in authentication** list box is unchecked.*

Removing a Target

To remove a target server from the list:

- 1 In the Proxy dialog, click the **Forward to** drop-down list and select the target server you'd like to remove.
- 2 Click **Remove**.

Steel-Belted Radius as a Target

This section describes how to set up proxy forwarding from some other RADIUS server (the proxy) to the Steel-Belted Radius server (the target):

- 1 Set up the proxy as a RADIUS client of Steel-Belted Radius.
Add the entry using the RAS Clients dialog. Specify the proxy's name, its IP address, and the shared secret that you want to use for encryption between the proxy and Steel-Belted Radius.
- 2 Ask the administrator at the target site to log into the proxy's RADIUS configuration program and set up Steel-Belted Radius as a Proxy RADIUS target. You'll need to provide this administrator with the IP address of the Steel-Belted Radius server.

Note: Make sure that the same UDP port and shared secret are entered on both proxy and target sides.

Dictionaries when Steel-Belted Radius is the Target

When Steel-Belted Radius receives a proxy-forwarded packet, it consults its RAS Client entry for that proxy server. The **Make/model** field of this entry determines which attribute dictionary Steel-Belted Radius uses.

At various different times, Steel-Belted Radius can receive requests from the same proxy server that have originated from different NAS devices, possibly of different types. The single **Make/model** field that was entered for the proxy might not be adequate to handle the variety of NASs on the "other side" of the transaction.

One way to handle this problem is to add the originating NAS devices to Steel-Belted Radius's list of RAS Clients. Steel-Belted Radius can be configured to examine each proxy-forwarded packet for clues as to the make and model of the originating device. If clues are found, Steel-Belted Radius does everything it can to map this information to a vendor-specific dictionary, and uses this dictionary in preference to the one for the proxy.

Accepting Packets from Any Proxy

If you'd like Steel-Belted Radius to be able to accept proxy requests from any IP address, you can use the RAS Clients dialog to add a special entry called `<ANY>`, and specify a shared secret. The `<ANY>` entry permits forwarded requests from any proxy to be accepted, provided the shared secret is correct.

Note: This feature requires that proxies are configured to use the shared secret you provide in the `<ANY>` entry.

Proxy RADIUS as an Authentication Method

Any target server can be configured as a Steel-Belted Radius authentication method. Simply enable the **Include in authentication** list option in the corresponding Proxy database entry.

A target server can be set up as an authentication method even if the end users don't know anything about the target. That is, a user does not need to log in using a decorated username such as `User@TargetName` to be authenticated by the target server.

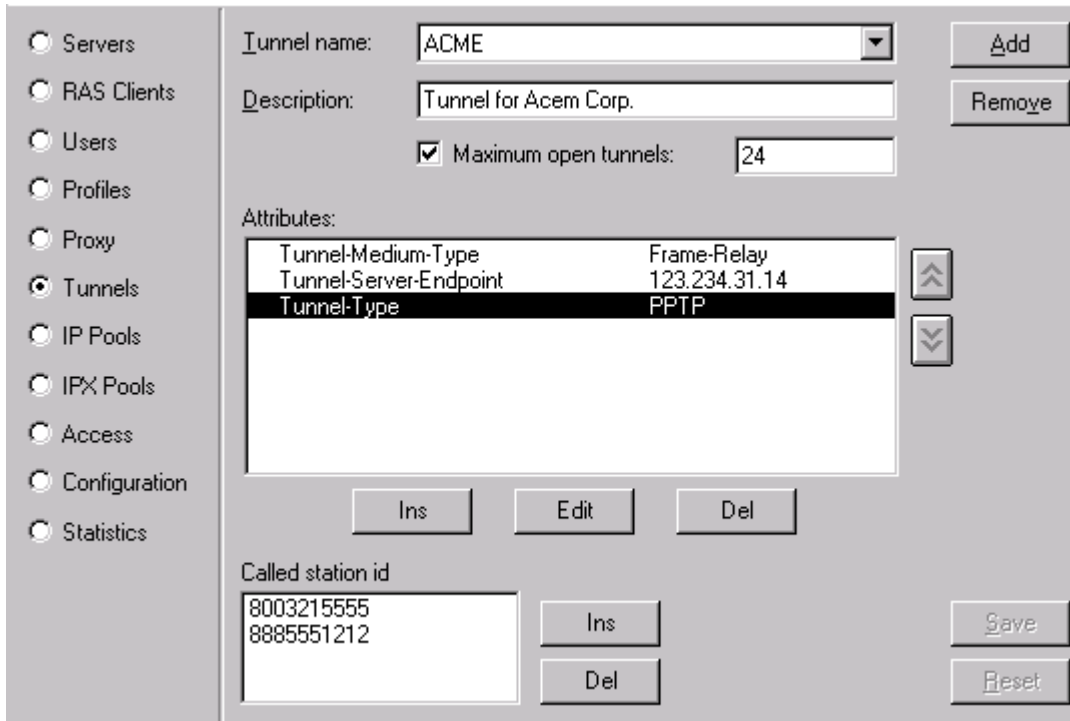
If you prioritize the `proxy: TargetName` authentication method above the Native User authentication method in the Authentication Methods list, the user can log in as `User` and Steel-Belted Radius automatically sends the request to the target for authentication. The authentication succeeds if the Username and password are stored on the target, but if not, Steel-Belted Radius reaches the Native User method eventually, and the user can then be authenticated.

This technique is useful as a migration path to Steel-Belted Radius from other RADIUS servers. You can set up Steel-Belted Radius as the proxy and the old RADIUS server as the target. After proxy authentication is enabled (in the Proxy dialog) and prioritized (in the Configuration dialog), Steel-Belted Radius can authenticate users against the old RADIUS server, either as an automatic “first choice” or as an alternative when authentication against the new server's “native” database fails.

Tunnels Dialog

The Tunnels dialog lets you configure Steel-Belted Radius to support tunnels. When you add a Tunnel entry, you're not creating a tunnel; you're enabling Steel-Belted Radius to support an existing tunnel's authentication and accounting needs.

See also “Tunnels” on page 72.



Tunnels Dialog (Windows version)

Adding a Tunnel

To add a Tunnel entry:

- 1 In the Tunnels dialog, click **Add**. The Add New Tunnel dialog appears.
- 2 Enter the Tunnel name and click **OK**.

Tunnel names do not need to match the actual node name of a client tunnel server. Tunnel name cannot duplicate any other target name, realm name, or tunnel name in your Steel-Belted Radius configuration.

The name you entered appears in the **Tunnel name** field, with blank settings beneath.

- 3 Edit the settings for the Tunnel entry. Be sure you fill in all required fields.
- 4 Click **Save** to make your changes permanent.

Editing a Tunnel

The settings for each Tunnel include the maximum number of concurrent connections using this tunnel, and any attributes that the RAS client needs to complete the tunnel connection.

You can configure these settings using the Tunnels dialog.

- 1 Set the **Maximum open tunnels** number.
See “Concurrent Tunnel Connections” on page 80.
- 2 Enter a text **Description** of the tunnel. This text is for administrative use only and does not affect tunnel connections.
- 3 Each Tunnel entry can provide various Attributes. At the time of connection, these attributes are filtered according to the Make/model of the RAS Client used to establish the connection.

You can add, modify, and remove Attributes in the Tunnels dialog using similar techniques as in other dialogs. Use the **Ins**, **Edit**, and **Del** buttons underneath the **Attributes** list box, and the up- and down-arrow buttons to the right of the **Attributes** list box.

See “Editing User Settings” on page 96, especially “Adding New Attributes” on page 96 and following.

- 4 To add a telephone number to the **Called station Id** list box, click the **Ins** button to the right of the list box. To delete a number, highlight it in the list and click **Del**.
See “Called Station Id” on page 74.
- 5 Click **Save** to make your changes permanent, **Reset** to undo them.

Removing a Tunnel

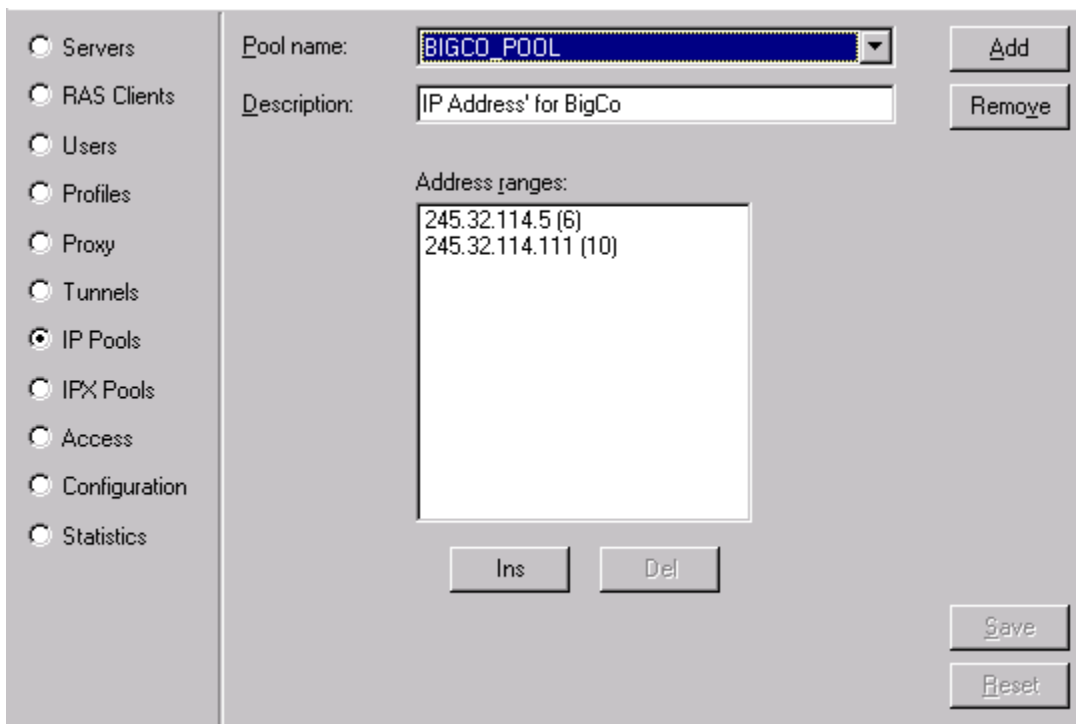
To remove a Tunnel entry from the Steel-Belted Radius database:

- 1 In the Tunnels dialog, click the **Name** drop-down list and select the Tunnel you would like to remove. (Under **UNIX**: Click **OK** to proceed.)
- 2 Click **Remove**. you are prompted to confirm the deletion.
- 3 Click **Yes**. The tunnel is removed from the list.

IP Pools Dialog

The IP Pools dialog allows you to set up one or more pools out of which unique IP addresses are assigned as users require them. Each pool consists of a list of one or more ranges of IP addresses.

Important: Depending on your overall configuration, certain limitations might apply to this feature. See “How Address Assignment Works” on page 77.



IP Pools Dialog (Windows version)

Adding an IP Address Pool

An IP address pool consists of one or more ranges of IP addresses. You can add or delete ranges and set an optional description for each address pool.

To add a new IP address pool:

- 1 In the IP Pools dialog, click **Add**. The Add New IP Address Pool dialog appears.

- 2 Enter the **Pool name** and click **OK**. The IP Pools dialog displays the name you entered in the **Pool name** field.
- 3 Enter a text **Description** of the address pool.
- 4 Each IP Pools entry can provide various **Address ranges**. Add and remove the ranges of IP addresses that make up the pool, as described in the following topic.
- 5 Click **Save** to make your changes permanent.

Editing an IP Address Pool

To add a new range of IP addresses to an IP address pool:

- 1 Select the pool from the **Pool name** drop-down list.
- 2 Click **Ins**. The Add New IP Address Range dialog appears.
- 3 Enter the starting address and the number of addresses in the new range, then click **Add**.
Repeat for as many address ranges as you'd like to add.
- 4 When done adding ranges, click **Close** to return to the IP Pools dialog.

To remove a range of addresses from an IP address pool:

- 1 Select the pool from the **Pool name** drop-down list.
- 2 Highlight the range in the **Address ranges** list.
- 3 Click **Del** (not **Remove**).

Removing an IP Address Pool

To remove an IP Pool entry from the Steel-Belted Radius database:

- 1 In the IP Pools dialog, click the **Pool name** drop-down list and select the IP Pool you would like to remove. (Under **UNIX**: Click **OK** to proceed.)
- 2 Click **Remove**. (Under **Windows**: you are prompted to confirm the deletion.)

Specifying IP Address Assignment in User/Profile Records

The Framed-IP-Address Return-List attribute controls how the server assigns an IP address to a user making a connection.

When you add or edit the Framed-IP-Address attribute in the Users or Profiles dialog, the Framed-IP-Address dialog appears.



Editing the Framed-IP-Address (Windows version)

This dialog allows you to select an IP address assignment option. Either:

- Type an IP address in the **Enter an IP address** field; *or*
- Check the **Assign IP address from pool** box and select the name of the pool from the list.

NAS-Specific IP Address Pools

Steel-Belted Radius allows you to define IP Address Pools that are specific to the NAS from which the user request was received. You can also define a set of suffixes that define categories of pools.

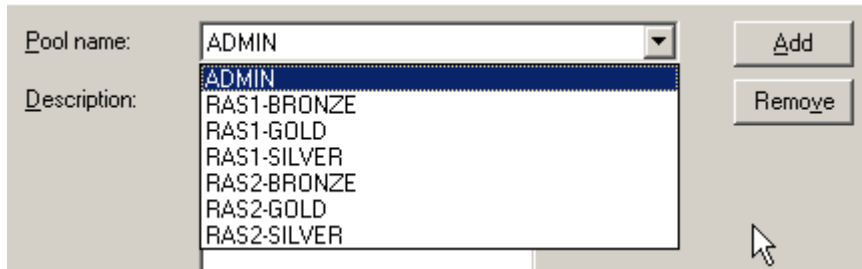
A pool category might correspond, for example, to the kinds of services available to users in that category. You might decide to define categories called `Bronze`, `Silver`, and `Gold`, indicating increasing packet routing priorities.

To create a NAS-specific address pool, you must follow these steps:

- 1 If you want NAS-Specific IP Address pools split into categories, define the appropriate suffixes in the [IPPoolSuffixes] section of `radius.ini`. For example:

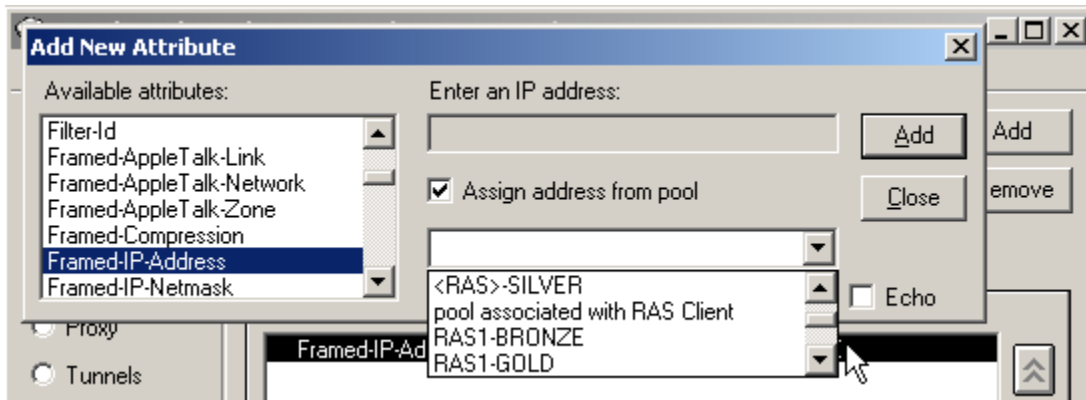
```
[IPPoolSuffixes]
-Bronze
-Silver
-Gold
```

- 2 Define the IP Address Pool with the IP Pools dialog.



IP Pools Dialog

- 3 Associate the new IP Address Pool with the appropriate NAS by use of **IP Address Pool** field on the RAS Clients dialog.
- 4 You can now assign a user to a NAS-Specific IP Address Pool and suffix. Create this association either with the Users dialog or the Profiles dialog.



Associating IP Address Pools with RAS Clients

See “radius.ini [IPPoolSuffixes] Section” on page 217.

If user Bob, who has been assigned to <RAS>-Bronze, logs into RAS1, he receives an IP address from the RAS1-Bronze address pool. If he logs into RAS2, he receives an address from the RAS2-Bronze address pool. If, however, he logs into RAS3 but RAS3-Bronze has not been defined in the IP Pools dialog, he is not assigned an IP address.

Specifying IP Address Assignment from a DHCP Server

IP addresses can be assigned from a backend DHCP server, rather than from a standard IP address pool. DHCP address pools function like internal address pools — Framed-IP-Address can be allocated from any address pool, either internal or DHCP.

DHCP address pools are defined in the `dhcp.ini` file and initialization files with the extension `.dhc`.

See “`dhcp.ini` File” on page 194.

In addition, each DHCP address pool must be enabled by adding a placeholder IP address pool in the Administrator. This placeholder pool should have the same name as the DHCP pool, and should have an empty list of address ranges. The placeholder pool allows the DHCP pool to appear in lists presented by the Administrator, so it can be selected into an attribute.

When an IP address must be assigned from a DHCP pool during an Access-Request, DHCP DISCOVER and REQUEST messages are issued to attempt the allocation of an address. When an accounting Stop ends the session, DHCP RELEASE is issued to the server that allocated the address. Upon receipt of an accounting INTERIM request, a DHCP REQUEST message is issued to the server that allocated the address, attempting to extend the lease. If the server is specified as a broadcast address, DHCP *failover* occurs if the primary DHCP server goes down.

DHCP leases can be acquired, extended, and released by different servers. The server that acquires the lease adds all the information for extending and releasing the lease to the Class attribute.

Flexible configuration features allow RADIUS attributes to be mapped to DHCP options. Therefore, information from a RADIUS request can be provided to the DHCP server, and information returned from the DHCP server can be returned to the NAS device.

During authentication, if an address is assigned from a pool, the pool name must refer to either a DHCP pool or an internal pool. If the pool name is not found, the request is rejected.

Address Allocation

During address allocation, a DISCOVER message is issued. If an OFFER is received from a DHCP server and the offered lease time meets the minimum lease time requirements, the server issues a REQUEST message. If an ACK message is received, the allocated address is returned in the Access-Accept.

In addition to the options required for normal DHCP operation, additional options in the DHCP DISCOVER and REQUEST messages are constructed based on the attributes in the RADIUS request and the literal values specified in the [Request] section for the pool. A Parameter Request List option is also constructed, listing all return options required for populating the RADIUS response, as specified in the [Reply] section for the pool.

If an address is assigned via DHCP, the `DH=` field is added to the Class attribute. This field includes:

- The unique client identifier for this lease.
- The address of the DHCP server.
- The lease time.

The unique client identifier for each user session is placed in the client hardware address field of the DHCP request as well as in the Client ID option. This information is used by the DHCP server to associate IP addresses with clients.

Address Renewal

If an INTERIM accounting message whose Class attribute includes both the IP= and the DH= fields is received, a REQUEST message is unicast to the DHCP server that allocated the address in an attempt to renew the lease. It requests the same lease time as was granted for the original request. If the server is specified as a broadcast address, DHCP *failover* occurs if the primary DHCP server goes down.

***Important:** If a renewal request is rejected, the DHCP server does not inform the NAS device that the user's IP address is not renewed and might become invalid. Address renewal occurs strictly on a "cross-your-fingers" basis.*

Address Release

If an accounting Stop message whose Class attribute includes both the IP= and the DH= fields is received, a RELEASE message is unicast to the DHCP server that allocated the address.

Note: The DHCP server does not reply to the RELEASE message.

An address to the DHCP server is also released when a session is deleted from its session database for reasons other than receiving an accounting Stop. For example, phantom session expiration or administrative deletion of a session result in the release of the address via DHCP.

DHCP Option Mapping

Options in a DHCP DISCOVER or REQUEST message can automatically be constructed based on attributes in the RADIUS request as well as pre-configured literal values. Also, options returned by the DHCP server in an OFFER message can be transmitted back to the NAS device in RADIUS attributes.

The following applies to the mapping between RADIUS attributes and DHCP options:

- Both standard and vendor-specific DHCP options are supported. (Vendor-specific DHCP options must use standard encapsulation rules, as described in RFC 2132.)
- Format conversions between RADIUS attributes and DHCP options are performed. For example, a DHCP option containing an IP address is formatted into dotted notation when returned in a RADIUS string attribute.
- A single RADIUS request attribute can set more than one DHCP options in a request, and a single DHCP option can set more than one RADIUS response attribute.
- A single DHCP option containing multiple values can be mapped to multiple instances of a single RADIUS attribute.

For example, a RADIUS attribute called `IP-Router` could appear multiple times in an Access-Accept. DHCP's Router option returns a list of IP addresses of routers. This single DHCP option can be configured to return multiple instances of the RADIUS IP-Router attribute -- one for each router address in the list.

- A single DHCP option containing multiple values can be mapped to multiple RADIUS attributes.

For example, two RADIUS attributes exist, `Primary-DNS-Server` and `Secondary-DNS-Server`. DHCP's DNS Server option returns a list of IP addresses of DNS servers. This single DHCP option can be configured to set the first DNS server address in `Primary-DNS-Server` and the second in `Secondary-DNS-Server`.

- Only attributes appropriate to the dictionary are returned.

Therefore, if NAS devices from different vendors use different RADIUS attributes for the same information, each RADIUS attribute that might be required can be mapped to the same DHCP option. The correct attribute is returned to the NAS device.

Using Multiple Servers

As the information required to renew or release a DHCP-assigned address is contained in the Class attribute, it is feasible to set up multiple servers, all utilizing a common DHCP server for address allocation. The NAS device can issue requests to any of the servers, and addresses are assigned and released correctly even if different servers handle authentication and accounting requests for the same session.

In order for this architecture to work correctly, each server must be configured to be stateless -- that is, the current sessions database must be turned off in the `radius.ini` file as follows:

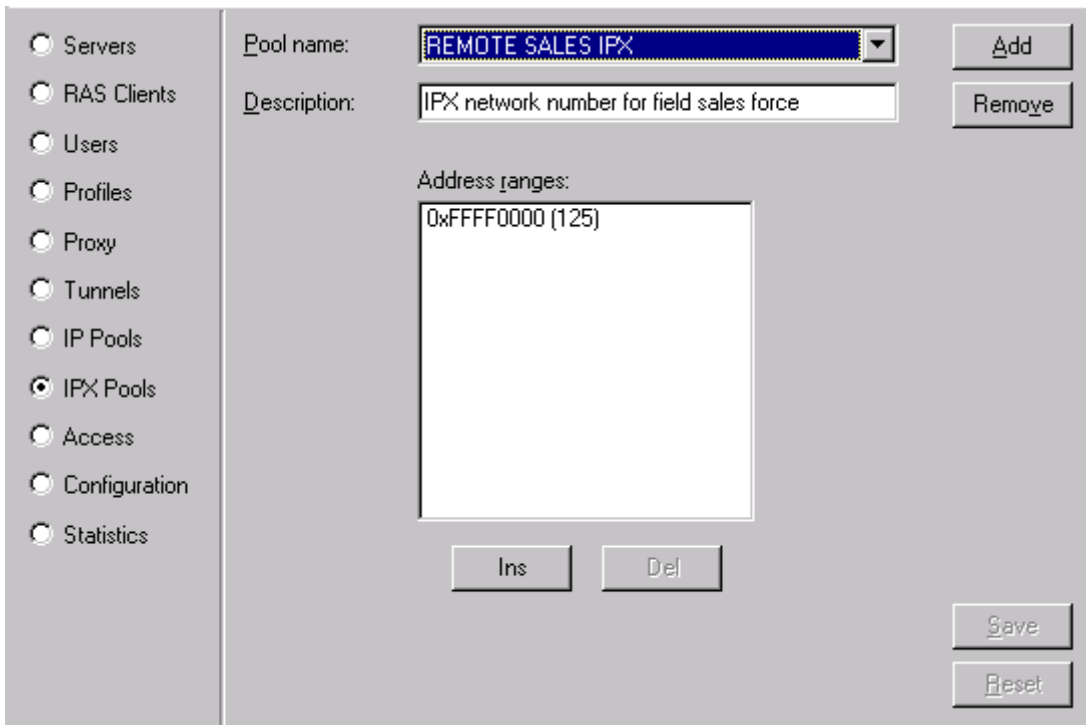
```
[CurrentSessions]
Enable = 0
```

Current sessions processing makes sense only when authentication and all accounting are directed to the same server. If current sessions processing is not disabled, the current sessions database is incorrect and always growing. For example, DHCP addresses are prematurely released when phantoms expire.

IPX Pools Dialog

The IPX Pools dialog allows you to set up one or more pools out of which unique IPX network numbers are assigned as users require them. Each pool consists of a list of one or more ranges of IPX network numbers.

Important: Depending on your overall configuration, certain limitations might apply to this feature. See “How Address Assignment Works” on page 77.



IPX Pools Dialog (Windows version)

Adding an IPX Pool

An IPX pool consists of one or more ranges of IPX network numbers. You can add or delete ranges and set an optional description for each address pool.

To add a new pool of IPX network numbers:

- 1 In the IPX Pools dialog, click **Add**. The Add New IPX Address Pool dialog appears.
- 2 Enter the **Pool name** and click **OK**. The IP Pools dialog displays the name you entered in the **Pool name** field.
- 3 Enter a text **Description** of the address pool.
- 4 Each IP Pools entry provides various address ranges. Add and remove the ranges of IP addresses that make up the pool, as described in the following topic.
- 5 Click **Save** to make your changes permanent.

Editing an IPX Pool

To add a new range of IPX network numbers to an IPX address pool:

- 1 Select the pool from the **Pool name** drop-down list.
- 2 Click **Ins**. The Add New IPX Address Range dialog appears.
- 3 Enter the starting IPX network number and the number of addresses in the new range, then click **Add**.
Repeat for as many address ranges as you'd like to add.
- 4 When done adding ranges, click **Close** to return to the IPX Pools dialog.

To remove a range of network numbers from an IPX address pool:

- 1 Select the pool from the **Pool name** drop-down list.
- 2 Highlight the range in the **Address ranges** list.
- 3 Click **Del** (not **Remove**).

Removing an IPX Pool

To remove an IPX Pool entry from the Steel-Belted Radius database:

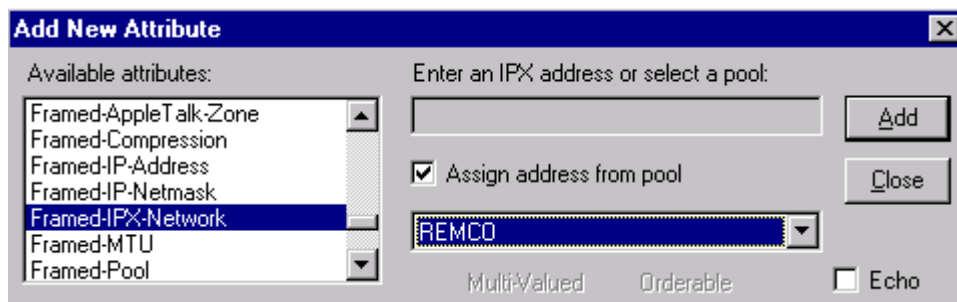
- 1 In the IPX Pools dialog, click the **Pool name** drop-down list and select the IPX Pool you would like to remove. (Under **UNIX**: Click **OK** to proceed.)
- 2 In the IPX Pools dialog, click **Remove**.

Specifying Pooled IPX Network Numbers in User/Profile Records

The Framed-IPX-Address Return-List attribute controls how the Steel-Belted Radius server assigns an IPX address to a user making a connection.

When you add or edit the Framed-IPX-Address attribute in the Users or Profiles dialog, the Framed-IPX-Address dialog appears. This dialog allows you to select an IPX address assignment option. Either:

- Type an IPX address in the **Enter an IPX address** field; *or*
- Check the **Assign IPX address from pool** box and select the name of the pool from the list.



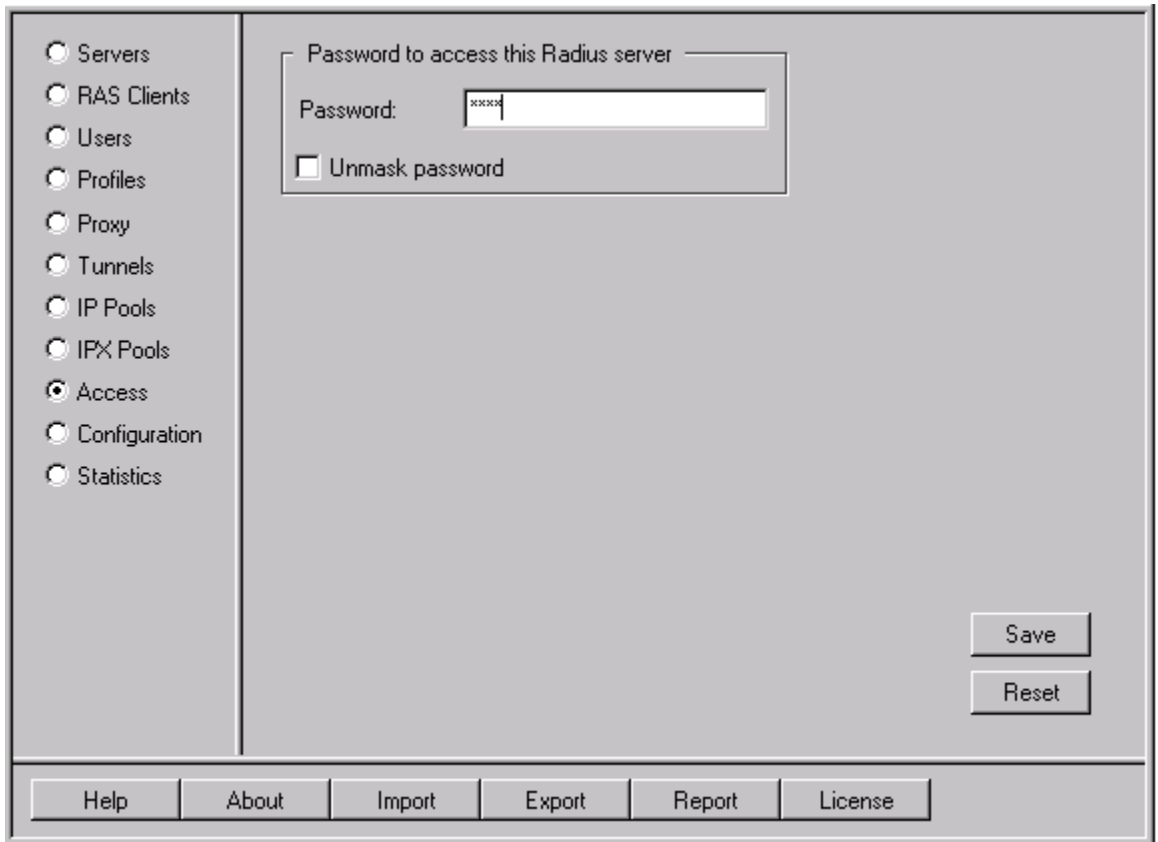
Specifying an IPX Pool for the Framed-IPX-Address Attribute (Windows version)

Access Dialog

UNIX Only

Each time you request a connection from the Servers dialog, the Administrator program prompts you to authenticate yourself by entering a Steel-Belted Radius administrative account name and password. If you enter the name of the default, full-service administrative account (`admin`), then you must enter the password defined in the Access dialog. Steel-Belted Radius is shipped with this password set

to radius. We suggest you change it immediately after installing the product. To do this, you must start the Administrator program and select the Access dialog.



Access Dialog (UNIX version)

If you want to control administrative access at a finer level of detail, Steel-Belted Radius allows you to designate other UNIX user and/or group accounts on the server as administrative accounts. You can also assign various levels of administrative privilege to these accounts.

See “access.ini File” on page 177 and “admin.ini File” on page 184.

Setting the Server Password

To set the password for the default administrative account (admin):

- 1 In the Servers dialog, select a server name.
- 2 Choose the Access dialog.

- 3 Enter a password in the **Password** field.
- 4 If you want to see the password characters, check the **Unmask password** checkbox. If you want the password concealed as asterisks (*****), uncheck the box.

Note: You do not have an opportunity to retype the password.

- 5 Click **Save** to keep your changes, **Reset** to undo them.
You can change this password again at any time.

Resetting the Server Password from the Default

If you forget the password for the default administrative account (`admin`), you can reset it to its default value and then change it to a value of your choice, as follows:

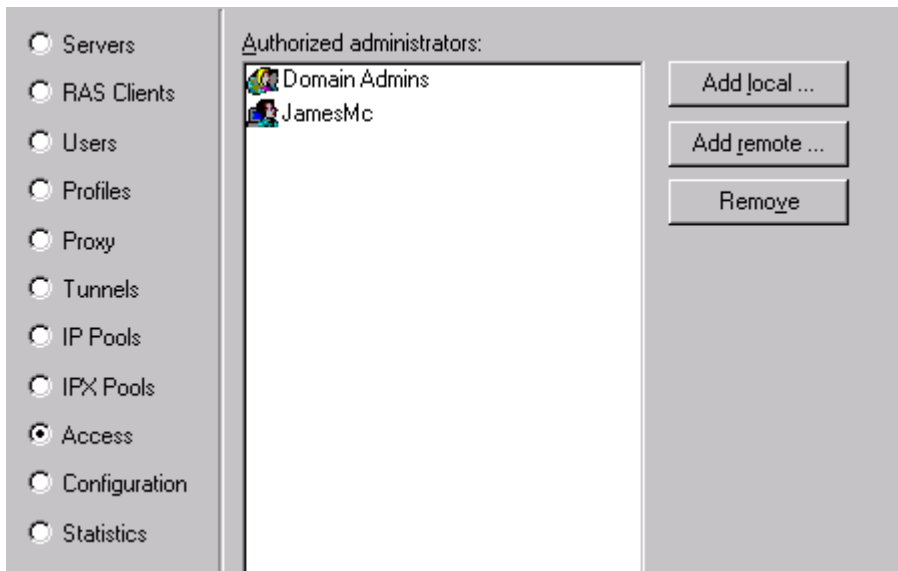
- 1 Create a new file (even an empty file is fine), name it `resetpwd`, and place it in the Steel-Belted Radius server directory that you defined at installation time.
- 2 Stop and restart the radius daemon.
- 3 Change the password from its installation default (`radius`).

Windows Only

The Access dialog lets you grant and revoke the right to use the Administrator program to configure a Steel-Belted Radius server.

When a Steel-Belted Radius server is first installed, any account that is a member of the NT group `Administrators` on a Steel-Belted Radius server implicitly has the right to use the Administrator program at its default (full) level of access. The Access dialog allows you to selectively grant and revoke the right to use the Administrator program, beginning from this starting point.

In the Access dialog, the RADIUS Administrators list show the users and groups that have been explicitly granted the right to run the Administrator. Local users or groups are shown with their normal name. Remote users or groups are shown with the name of the Domain, followed by a backslash and then the name of the Domain user or group.



Access Dialog (Windows version)

If you want to control administrative access at a finer level of detail, Steel-Belted Radius allows you to designate other NT user and/or group accounts as administrative accounts. You can also assign various levels of administrative privilege to these accounts.

See “access.ini File” on page 177 and “admin.ini File” on page 184.

Adding a Local Administrator

To grant access to a local administrator:

- 1 Click **Add local**.
- 2 A list appears, allowing you to select which users or groups should have Administrator access rights.
- 3 Select a user or group from the list. Click **Add**. Continue until you are done, and then click **Close**.

To revoke rights, highlight the user or group whose administration rights you'd like to revoke, and click **Remove** in the main Access dialog.

Adding a Remote Administrator

To grant access to a remote administrator within a Domain:

- 1 Click **Add remote**.

- 2 A list of Domains appears. Select a Domain name within which you would like to grant access.
- 3 Select a user or group from the list. Click **Add**. Continue until you are done, and then click **Close**.

To revoke rights, highlight the user or group whose administration rights you'd like to revoke, and click **Remove**.

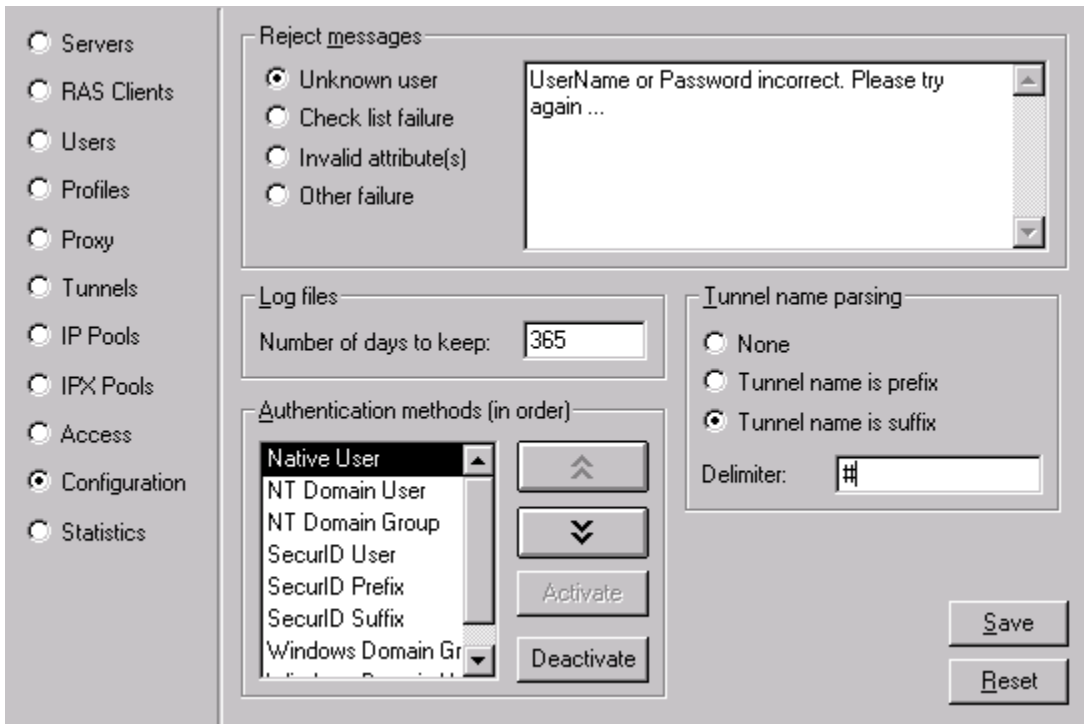
***Important:** Be careful not to revoke your own rights. If you do so, you will no longer have access to RADIUS administrative functions.*

Configuration Dialog

The Configuration dialog permits you to define how Steel-Belted Radius performs authentication and accounting. You can configure:

- The order in which different authentication methods are attempted.
- The number of days that the RADIUS server should retain event, authentication and accounting logs before the files are recycled.
- The text of messages sent to the NAS (and possibly to the User) when a RADIUS request is rejected.

- How the server should parse tunnel names; that is, should the tunnel name or user name be first, and which character should separate the names?



Configuration Dialog (Windows version)

You can edit information in any of the fields that appear on the screen. To make any changes you have entered permanent, click **Save**, and the new settings take effect. To revert to the previous settings, click **Reset**.

Reject Messages

When issuing an Access-Reject, Steel-Belted Radius can indicate the reason why the request was rejected. You can configure the message text returned to the client when a particular type of error occurs. This text is inserted into the standard RADIUS attribute Reply-Message within the Access-Reject response.

The following table lists the errors to which you can assign message text, and the meaning of each error.

Error	Meaning
Unknown User	The username and password authentication failed.
Check List Failure	The user was authenticated but is being rejected because the RADIUS request did not fulfill the requirements of the Check-List.
Invalid Attribute(s)	The request contained an attribute in violation of the RADIUS specification.
Other	Some other error occurred, such as a resource failure.

To modify message text:

- 1 In the Configuration dialog **Reject messages** panel, select an error type. The current message text displays to the right.
- 2 In the **message text display** field, edit the current text, or type a new message.

Tunnel Name Parsing

Use the fields in the Tunnel Name Parsing panel to configure a parsing convention for all of the tunnels that use the Steel-Belted Radius server to support RADIUS authentication and accounting. You can set the following options in the Configuration dialog:

- Choose **Tunnel name is suffix** to parse names as *User<Delimiter>TunnelName*.
- Choose **Tunnel name is prefix** to parse names as *TunnelName<Delimiter>User*.
- Choose **None** to disable tunnel name parsing.

If you choose this option, the tunnel authentication sequence is bypassed for each Access-Request; the server uses the standard username/password authentication sequence only.

- You can choose a *<Delimiter>* character other than '@' (the default).

You must set these options for a server. You cannot set these options for individual tunnels.

Note: You should choose different delimiter characters and different prefix/suffix name parsing conventions for Tunnels and for Proxies or realms. See “User-Names with a Single Delimiter” on page 60. See also “proxy.ini [Configuration] Section” on page 272.



Authentication Methods Configuration

When Steel-Belted Radius receives a user name, it does not know in advance to which authentication category this user belongs. It must try each method that it currently has configured and enabled. The Authentication Methods list allows you to fine-tune the sequence of authentication attempts.

See “Configuring Authentication Methods” on page 38.

To change the order in which the methods are tried:

- 1 Highlight the method in the list box.
- 2 Click one of the arrow buttons as follows:

Button	Action
	Moves the selected method up one slot in the list. If the selected method is already first in the list, then the button is disabled.
	Moves the selected method down one slot in the list. If the selected method is already last in the list, then the button is disabled.

To remove a method from the search list entirely:

- 1 Highlight the method in the list box.
- 2 Click the **Deactivate** button.

Log Files

Steel-Belted Radius records all transactions to log files. There are separate logs for authentication transactions and accounting attributes. For more information, see “Radius Log File” on page 144, “Authentication Log File” on page 146, and “Accounting Log File” on page 148.

Each day at midnight, the previous day’s log files are completed, and new log files are created for the new day’s transactions. To prevent the log files from continuously depleting available disk space Steel-Belted Radius retains the log files for a specified period of time and then automatically deletes them. To specify the number of days to retain log files, enter a value in the **Days to Keep** field.

Import/Export Capabilities

Steel-Belted Radius's Import/Export feature lets you export database information from any Steel-Belted Radius server and import it into another. This gives you a head start if you are configuring multiple servers.

Import and Export are selective; that is, you are given the opportunity to select exactly which items to export or import.

Steel-Belted Radius uses a specially formatted text file called a RADIUS Information File (.rif) for export and import.

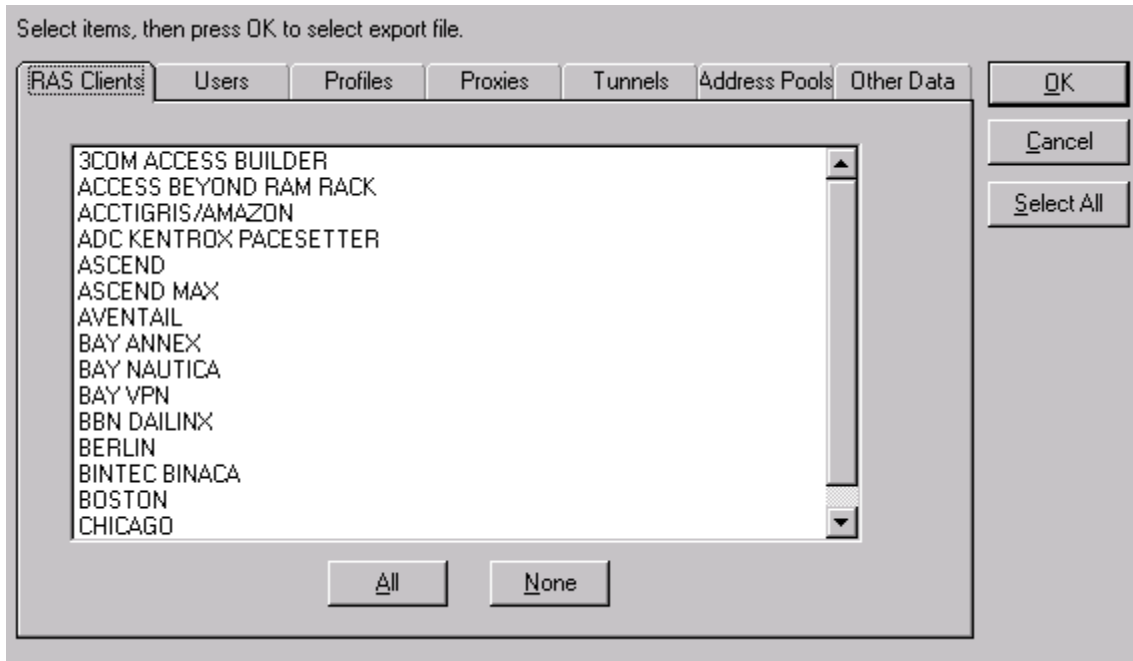
In addition to the native .rif format, Steel-Belted Radius permits importing of user data from the file format used in older, freeware implementations of the RADIUS standard, commonly deployed on UNIX systems. Different vendors' variations of this file format are supported via dictionaries.

Exporting to a RADIUS Information File

To export Steel-Belted Radius database information to a RADIUS Information File:

- 1 Run the Administrator program.
- 2 Depending on your platform:
 - Under **Windows**: Select the **File Export** command.
 - Under **UNIX**: Click the **Export** button that appears at the bottom of the Administrator display.

A dialog appears. Each tab in the dialog lists items of a particular category that you can export.



Export Dialog (Windows version)

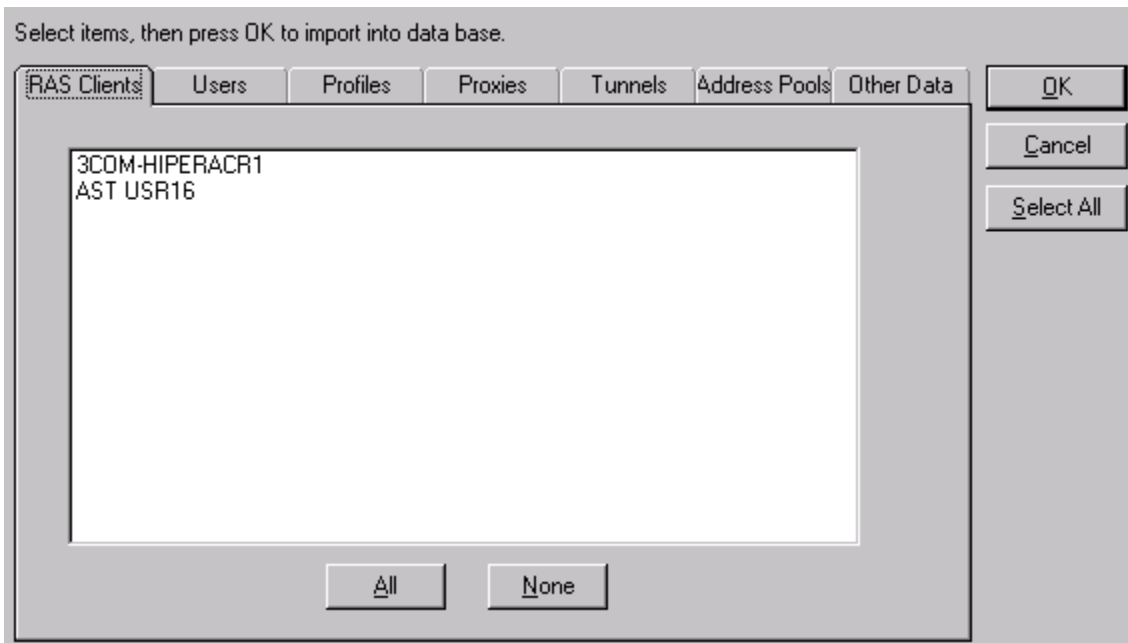
- 3 For each category, select the appropriate tab and click each item you'd like to export. To select all items in the category, click **All**.
To select all items in all categories, click **Select All**.
- 4 After you've selected all the items you want, click **OK**.
- 5 Depending on your platform:
 - Under **Windows**: A file browsing dialog appears. Specify an export file and click **Save**.
 - Under **UNIX**: The Select File dialog appears. Specify the full pathname of an export file and click **OK**.

Importing from a RADIUS Information File

To import from a RADIUS Information File into your Steel-Belted Radius database:

- 1 Run the Administrator program.
- 2 Depending on your platform:

- Under **Windows**: Select the **File Import** command. A file browsing dialog appears. Make sure the file type indicates RADIUS Information File (*.rif). Select an import file and click **Open**.
 - Under **UNIX**: Click the **Import** button that appears at the bottom of the Administrator display. The Select File dialog appears. Specify the full pathname of an export file and click **OK**.
- 3 The Import (or, under UNIX, the Import/Export) dialog appears, with each tab listing items of a particular category that are available for import from the selected file.



Import Dialog (Windows version)

- 4 For each category, select the appropriate tab and highlight each item you'd like to import. To select all items in the category, click **All**.
To select all items in all categories, click **Select All**.
- 5 After you select all the items you want, click **OK**. The items you selected are added to the Steel-Belted Radius database.
If an import item already exists, you are given the opportunity to confirm that you want to replace the existing entry with the entry from the import file.

Importing from Other File Formats

There are many RADIUS implementations currently deployed, mostly UNIX systems that are based on source code from Livingston Enterprises and Ascend. These implementations store data in a specially formatted text file normally called **users**.

To import user data from a **users** file into your Steel-Belted Radius database:

- 1 Run the Administrator program.
- 2 Depending on your platform:
 - Under **Windows**: Select the **File Import** command. A file browsing dialog appears. Select a file type of **External User Data File (*.*)**. Select an import file and click **Open**.
 - Under **UNIX**: Click the **Import** button that appears at the bottom of the Administrator display. The Select File dialog appears. Specify the full pathname of the **users** file and click **OK**.
- 3 The Import Options dialog appears. Modify as desired:
 - Set **File format** based on the NAS that the originating RADIUS server was meant to work with. This allows Steel-Belted Radius to correctly interpret the attribute naming conventions of a particular vendor.
 - If you check **Ignore attributes**, only names and passwords are imported. Check-List and Return-List attributes present in the imported file are excluded from the copied entries.
 - Select the Steel-Belted Radius profile to be applied to each imported user entry, or leave the entry set to **<no profile>**. If you do select a profile, ensure that **Ignore attributes** is checked.
 - Select **Allow PAP or CHAP** or **Allow PAP only** depending on how you'd like to store passwords in the database.
 - When you are satisfied with the settings, click **OK**.
- 4 The Import dialog appears, showing each user entry available for import from the selected file.

Highlight each user you'd like to import. To select all users, click **All**. After you select all the users you want, click **OK**.

The users you selected are added to the Steel-Belted Radius database as Native users. If a user is already present in the database, you are asked to confirm that you want to replace the existing user with the new user entry from the import file.

When you want to import user data from a users file into your Steel-Belted Radius database, and the users file contains attributes that you want to be imported along with the usernames and passwords, it is important to note the following:

- You must have at least one RAS Client entered in the Steel-Belted Radius Administrator. This allows a dictionary file of some type to be associated with the Steel-Belted Radius database, which in turn allows the attributes being imported from the users file to be recognized by the Steel-Belted Radius database.
- Even with a RAS Client entered, all of the attributes from the users file might not be recognized by Steel-Belted Radius due to the different versions of these RADIUS implementations and the fact that attribute names might be different in each.

To help with these differences, Steel-Belted Radius includes three dictionary import files. These files reside in your Steel-Belted Radius server directory and have a .dci extension. The files are named `annex.dci`, `ascend.dci`, and `portmstr.dci`. One of these three files should be a close match to your users file.

When attributes are involved, the best approach to importing from other file formats is the following:

- 1 Add a RAS Client in the Steel-Belted Radius Administrator.
- 2 Review the users file to see what attributes are associated with the users and compare them with the attribute names in the appropriate dictionary import file.

For example, if the users file from a Livingston Portmaster contains the attribute `User-Service-Type`, the `portmstr.dci` file in your server directory indicates the same attribute is named `Service-Type`. This difference in name would cause the import process to log an error and after the import was complete, the attribute in question would not be associated with the appropriate user(s). To avoid this, edit your `portmstr.dci` file so that `Service-Type` is globally changed to `User-Service-Type`. When the attribute name in the `portmstr.dci` file matches the attribute name in the users file, the file import can proceed without errors. Repeat this for any attribute names that do not match.

- 3 After Step 2 is completed, proceed with a normal import from the Steel-Belted Radius Administrator (as described above). Be sure to select the appropriate dictionary import file type when prompted at the Select Import Options dialog.

Logging, Monitoring, and Reporting

5

- A Window on Operations
- Radius Log File
- Authentication Log File
- Accounting Log File
- Statistics Dialog
- Sessions List
- Reporting Capabilities
- Windows NT Performance Monitor
- Windows NT Events

A Window on Operations

Steel-Belted Radius provides a variety of diagnostic features:

- Event, authentication, and accounting log files record the details of every RADIUS transaction on the Steel-Belted Radius server.
- The Administrator program's Statistics dialog, SNMP, and the LDAP Configuration Interface allow you to quickly view ongoing counts of the most significant statistics relating to authentication, accounting, and Proxy RADIUS transactions on the Steel-Belted Radius server.
- The Sessions list provides a record of all currently active sessions that were authenticated by this Steel-Belted Radius server.
- Tabular reports allow you to view selected contents of the Steel-Belted Radius database in written form
- **Windows only:** Windows NT Performance Monitor (perfmon) counters let you collect and interpret statistics about the Steel-Belted Radius service.
- **Windows only:** Windows NT events help you detect and solve system-related problems with the service.
- **UNIX only:** Steel-Belted Radius can report its activities to an SNMP Master Agent.

Radius Log File

Each time a RADIUS event occurs, it is recorded in the radius log file. The following are typical log entries:

- Sent accept response for user USERNAME to client RAS-Client-Name
- Unable to find user USERNAME with matching password
- Sent reject response
- Shutting down RADIUS Authentication Server ...
- Starting RADIUS Authentication Server ...

Radius log files are in ASCII format, and are intended for viewing by the network administrator. Each line of the radius log file contains a line with the date and time, followed by event information.

Radius log files are located in the RADIUS database directory area by default, although you can specify an alternate destination directory in the [Configuration] section of radius.ini. Radius log files are named *yyyymmdd.log*, where *yyyy* is the 4-digit year, *mm* is the month, and *dd* is the day on which the log file was created.

Radius log files are kept for the number of days specified in the Configuration dialog (described on page 134). After that time, older log files are deleted from the server to conserve disk space.

You can open the current log file while Steel-Belted Radius is running.

Level of Logging Detail

You can control the level of detail recorded in radius log files by use of the LogLevel, LogAccept, and LogReject settings.

The LogLevel setting determines the level of detail given in the radius log file. The LogLevel can be the number 0, 1, or 2, where 0 is the least amount of information, 1 is intermediate, and 2 is the most verbose. It is specified in the [Configuration] section of radius.ini and in the [Settings] sections of .aut files.

The LogAccept and LogReject flags allow you to turn on or off the logging of Access-Accept and Access-Reject messages in the radius log file. These flags are set in the [Configuration] section of radius.ini: a value of 1 (the default) causes these messages to be logged, and a value of 0 causes the messages to be omitted. An Accept or Reject is logged only if LogAccept or LogReject, respectively, is enabled *and* the LogLevel is “verbose” enough for the message to be recorded.

The TraceLevel setting specifies whether packets should be logged when they are received and being processed, and what level of detail should be recorded in the log.

If you alter the LogLevel or TraceLevel settings, you can have them take effect without restarting the server by issuing the following command:

- Under **UNIX**:

kill -HUP *pid*

where *pid* is the process id of your Steel-Belted Radius server.

- Under **Windows**:

radhup

Authentication Log File

The authentication log file records each RADIUS authentication request received by the Steel-Belted Radius server. Authentication log files are Comma Separated Value (CSV) ASCII text files that can be imported into a spreadsheet or database program.

Authentication log files are located in the RADIUS database directory area by default, although you can specify an alternate destination directory in the [Configuration] section of `authlog.ini`. Authentication log files are named `yyyymmdd.authlog`, where `yyyy` is the 4-digit year, `mm` is the month, and `dd` is the day on which the log file was created.

Authentication log files are kept for the number of days specified in the Configuration dialog. After that time, older log files are deleted from the server to conserve disk space.

The current log file can be opened while Steel-Belted Radius is running.

Authentication Log File Format

The first five fields in every authentication log entry are required by Steel-Belted Radius:

- `Date` – The date when the event occurred
- `Time` – The time when the event occurred
- `RAS-Client` – The name or IP address of the RAS Client sending the authentication record
- `Full-Name` – The fully distinguished name of the user, based on the authentication performed by the RADIUS server
- `ACC/REJ` – The result of the authentication request (ACCEPT or REJECT)

The RADIUS attributes specified in the `authlog.ini` file appear next. Attributes in the `authlog.ini` file beginning with a semicolon (;), are commented out, and their values are not recorded in the authentication log file.

```
User-Name
NAS-IP-Address
NAS-Port
Service-Type
Framed-Protocol
Framed-IP-Address
Framed-IP-Netmask
Framed-Compression
Login-IP-Host
```

```
Callback-Number
State
Called-Station-Id=
Calling-Station-Id=
NAS-Identifier=
Proxy-State=
Login-LAT-Service
Login-LAT-Node
Login-LAT-Group
Event-Timestamp
NAS-Port-Type
Port-Limit
Login-LAT-Port
```

Note: If the User-Password attribute is included in the authlog.ini file, it is ignored during processing to prevent exposure of users' cleartext passwords in the log file.

You can include vendor-specific attributes if the device sending the authentication packet supports them. For more information, see “Vendor-Specific Attributes” on page 53.

You can edit the authlog.ini file to add, remove or reorder the standard RADIUS or vendor-specific attributes that are logged. For more information, see “authlog.ini File” on page 186.

First Line Headings

The first line of the authentication log file lists the names of all the attributes that have been enabled for logging, in the order in which they are logged. This first line serves as a complete set of column headings for the remaining entries in the file. The content of the first line depends on the attributes specified in the authlog.ini file.

The following example shows the heading line and an authentication log file entry consisting of the required attributes.

```
"Date", "Time", "RAS-Client", "Full-Name", "ACC/REJ"
"7/3/2003", "12:11:55", "RRAS", "EdisonCarter", "ACCEPT",
```

Comma Placeholders

It's possible that not all the attributes expected in the first line of the authentication log file have had data returned for them by the currently logged event. If this is the case, when Steel-Belted Radius writes the event to the authentication log file, it uses a comma “placeholder” to mark the location of each empty entry, so that all entries remain correctly aligned with their headings.

For example, the following log entries indicate that Bob's authentication request was rejected but Alice's authentication request was accepted. The reported fields include Called-Station-Id, Calling-Station-Id, and Port-Limit. Note that the attributes listed in the log heading which were not returned for the authentication events are separated with commas.

```
"Date", "Time", "RAS-Client", "Full-Name", "Acc/Rej", "User-Name", "NAS-IP-Address",  
"NAS-Port", "Service-Type", "Framed-Protocol", "Framed-IP-Address", "Framed-IP-Net-  
mask", "Framed-Compression", "Login-IP-Host", "Callback-Number", "State",  
"Called-Station-Id", "Calling-Station-Id", "NAS-Identifier", "Proxy-State",  
"Event-Timestamp", "NAS-Port-Type", "Port-Limit", "Login-LAT-Port"  
"07/14/2003", "13:39:10", "192.168.2.42", "BOB", "REJECT",,,,,,,,,, "Alice's  
Office", "Bob's Office",,,,,,"5",  
"07/14/2003", "13:43:26", "192.168.2.42", "ALICE", "ACCEPT",,,,,,,,,, "Bob's  
Office", "Alice's Office",,,,,,"5",
```

Accounting Log File

RADIUS accounting events are recorded in the accounting log file. Accounting events include:

- START messages, indicating the beginning of a connection.
- STOP messages, indicating the ending of a connection.
- INTERIM messages, sent at regular intervals from a NAS to indicate that a user connection is still active.

Accounting log files use comma-delimited, ASCII format, and are intended for import into a spreadsheet or database program. Log file format is as described below.

Accounting log files are located in the RADIUS database directory area by default, although you can specify an alternate destination directory in the [Configuration] section of `account.ini`. Accounting log files are named `yyyymmdd.ACT`, where `yyyy` is the 4-digit year, `mm` is the month, and `dd` is the day on which the log file was created.

Accounting log files are kept for the number of days specified in the Configuration dialog. After that time, older log files are deleted from the server to conserve disk space.

The current log file can be opened while Steel-Belted Radius is running.

Accounting Log File Format

The first six fields in every accounting log entry are provided by Steel-Belted Radius for your convenience in reading and sorting the file:

- `Date` - the date when the event occurred
- `Time` - the time when the event occurred
- `RAS-Client` - the name or IP address of the RAS Client sending the accounting record
- `Record-Type` - START, STOP, INTERIM, ON, or OFF, the standard RADIUS accounting packet types
- `Full-Name` - the fully distinguished name of the user, based on the authentication performed by the RADIUS server
- `Auth-Type` - a number that indicates the class of authentication performed:
 - 0 - Native
 - 6 - Windows NT Domain User
 - 7 - Windows NT Domain Group
 - 8 - Windows NT Host User
 - 9 - Windows NT Host Group
 - 10 - SecurID User
 - 11 - SecurID Prefix
 - 12 - SecurID Suffix
 - 13 - UNIX User
 - 14 - UNIX Group
 - 15 - TACACS+ User
 - 16 - TACACS+ Prefix
 - 17 - TACACS+ Suffix
 - 100 - Tunnel User
 - 200 - External Database
 - (other) - Proxy

By default, all of the standard RADIUS attributes appear next. See “Standard RADIUS Accounting Attributes” on page 151.

You can include vendor-specific attributes if the device sending the accounting packet supports them. For more information, see “Vendor-Specific Attributes” on page 53.

You can edit the `account.ini` initialization file to add, remove or reorder the standard RADIUS or vendor-specific attributes that are logged. For more information, see “`account.ini` File” on page 178.

First Line Headings

The first line of the accounting log file lists the names of all the attributes that have been enabled for logging, in the order in which they'll be logged. This first line serves as a complete set of column headings for the remaining entries in the file.

The following example of a first line shows required headings in bold italic, standard RADIUS headings in bold, and vendor-specific headings in regular text:

```
"Date", "Time", "RAS-Client", "Record-Type", "Full-Name",  
"Auth-Type", "User-Name", "NAS-Port", "Acct-Status-Type",  
"Acct-Delay-Time", "Acct-Input-Octets", "Acct-Output-Octets",  
"Acct-Session-Id", "Acct-Authentic", "Acct-Session-Time",  
"Acct-Input-Packets", "Acct-Output-Packets",  
"Acct-Termination-Cause", "Acct-Multi-Session-Id",  
"Acct-Link-Count", "Acc-Err-Message",  
"Nautica-Acct-SessionId", "Nautica-Acct-Direction",  
"Nautica-Acct-CauseProtocol", "Nautica-Acct-CauseSource",  
"Telebit-Accounting-Info", "Last-Number-Dialed-Out",  
"Last-Number-Dialed-In-DNIS", "Last-Callers-Number-ANI",  
"Channel", "Event-Id", "Event-Date-Time",  
"Call-Start-Date-Time", "Call-End-Date-Time",  
"Default-DTE-Data-Rate", "Initial-Rx-Link-Data-Rate",  
"Final-Rx-Link-Data-Rate", "Initial-Tx-Link-Data-Rate",  
"Final-Tx-Link-Data-Rate", "Sync-Async-Mode",  
"Originate-Answer-Mode", "Modulation-Type",  
"Equalization-Type", "Fallback-Enabled", "Characters-Sent",  
"Characters-Received", "Blocks-Sent", "Blocks-Received",  
"Blocks-Resent", "Retrans-Requested", "Retrans-Granted",  
"Line-Reversals", "Number-Of-Characters-Lost",  
"Number-of-Blers", "Number-of-Link-Timeouts",  
"Number-of-Fallbacks", "Number-of-Upshifts",  
"Number-of-Link-NAKs", "Back-Channel-Data-Rate",  
"Simplified-MNP-Levels", "Simplified-V42bis-Usage",  
"PW_VPN_ID"
```

Comma Placeholders

It's possible that not all the attributes expected in the first line of the accounting log file have had data returned for them by the currently logged event. If this is the case, when Steel-Belted Radius writes the event to the accounting log file, it uses a comma "placeholder" to mark the location of each empty entry, so that all entries remain correctly aligned with their headings.

For example, based on the "first line" of headings described above, the following is a valid accounting log entry, in which the value of the Acct-Status-Type attribute is 7:

```
"12/23/1997", "12:11:55", "RRAS", "Accounting-On",
,,,,,7,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
```

Standard RADIUS Accounting Attributes

RFC 2866, "RADIUS Accounting," identifies standard RADIUS accounting attributes.

User-Name	The name of the user as received by the client.
NAS-Port	The port number on the client device.
Acct-Status-Type	A number that indicates the beginning or ending of the user service: 1 - Start 2 - Stop 3 - Interim-Acct 7 - Accounting-On 8 - Accounting-Off
Acct-Delay-Time	Indicates how many seconds the client has been trying to send this record, which can be subtracted from the time of arrival on the server to find the approximate time of the event generating this request.
Acct-Input-Octets	Number of octets (bytes) received by the port over the connection; present only in STOP records.
Acct-Output-Octets	Number of octets (bytes) sent by the port over the connection; present only in STOP records.
Acct-Session-Id	Identifier used to match START and STOP records in a log file.
Acct-Authentic	indicates how the user was authenticated by RADIUS, the RAS itself, or another remote authentication protocol: 1 - RADIUS 2 - Local 3 - Remote
Acct-Session-Time	Elapsed time of connection in seconds; present only in STOP records.
Acct-Input-Packets	Number of packets received by the port over the connection; present only in STOP records.
Acct-Output-Packets	Number of packets sent by the port over the connection; present only in STOP records.

Acct-Termination-Cause	Number that indicates how the session was terminated; present only in STOP records: 1 - User Request 2 - Lost Carrier 3 - Lost Service 4 - Idle Timeout 5 - Session Timeout 6 - Admin Reset 7 - Admin Reboot 8 - Port Error 9 - NAS Error 10 - NAS Request 11 - NAS Reboot 12 - Port Unneeded 13 - Port Preempted 14 - Port Suspended 15 - Service Unavailable 16 - Callback 17 - User Error 18 - Host Request
Acct-Multi-Session-Id	Unique accounting identifier to make it easy to link together multiple related sessions in a log file.
Acct-Link-Count	The count of links that are known to have been in a given multi-link session at the time the accounting record is generated.

Statistics Dialog

Steel-Belted Radius provides information on the status of the server. The Statistics dialog provides three tabs with statistics for all types of RADIUS activity on the currently selected server: Authentication, Accounting, and Proxy forwarding.

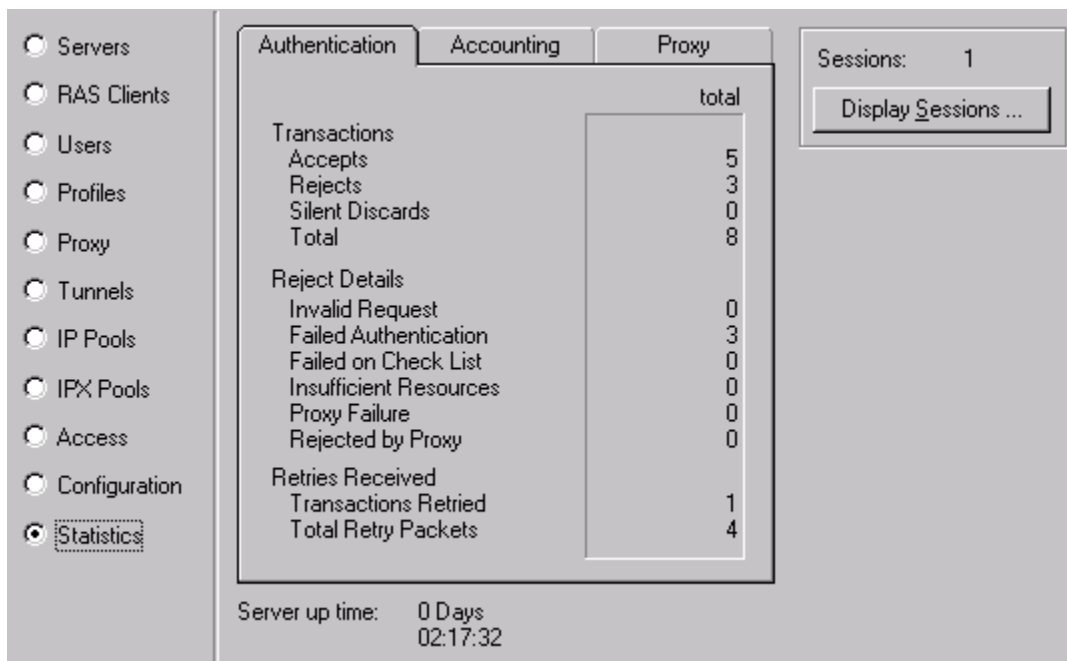
You can also:

- View the amount of time Steel-Belted Radius has been running.
- View an on-screen report of all users currently connected via a NAS or Tunnel, based on real-time RADIUS accounting information.

Note: Only the standard IETF RADIUS statistics are available from the Statistics dialog. To access Steel-Belted Radius extended statistics, you must use other utilities. See “Statistics Variables (LCI Only)” on page 357.

Authentication Statistics

Authentication statistics provide information such as the number of accept and reject messages and the reasons for rejecting authentication.



Statistics Dialog, Authentication Tab (Windows version)

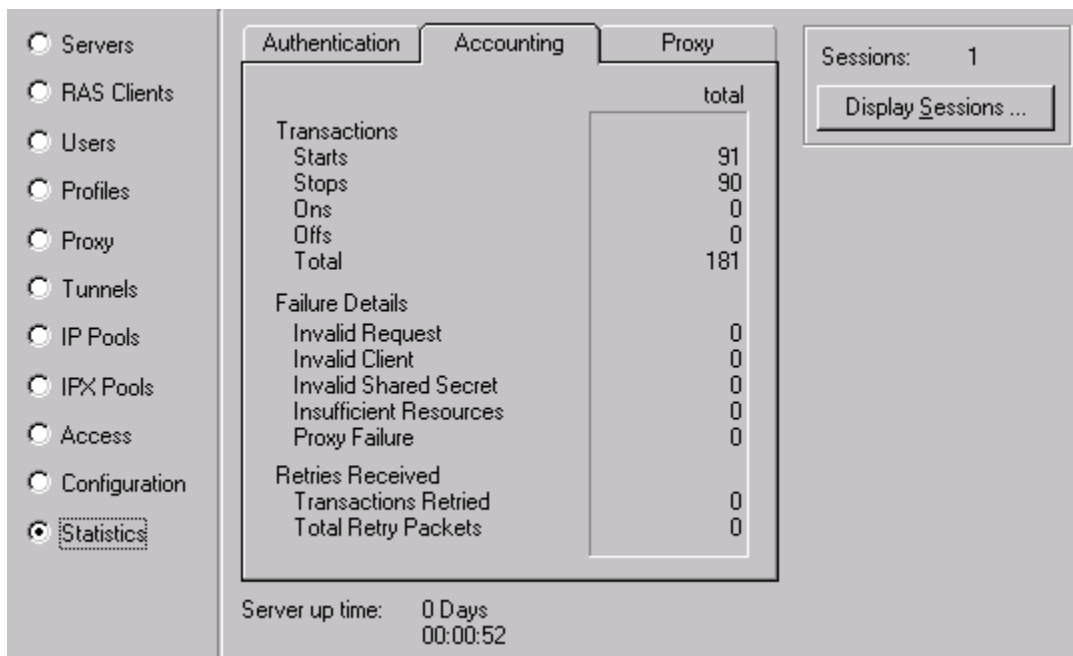
The following table describes the authentication statistics, with possible interpretations in italics.

Authentication	
Statistic	Meaning
Transactions	
Accepts	The total number of RADIUS transactions that resulted in an accept response.
Rejects	The total number of RADIUS transactions that resulted in a reject response. These are broken out in Reject Details below.
Silent Discards	The total number of requests in which the client could not be identified. <i>A device might be configured to use Steel-Belted Radius but no RAS Client entry has been created on the server with the name and/or IP address of the client; or the RAS Client entry might be configured with an incorrect name or IP address; or some rogue device is attempting to compromise RADIUS security.</i>

Authentication Statistic	Meaning
Total	The total of the three fields above.
Reject Details	
Invalid Request	The total number of invalid RADIUS requests made. <i>A device is sending incorrectly formed packets to Steel-Belted Radius; either there is a configuration error or the device does not conform to the RADIUS standard.</i>
Failed Authentication	The total number of failed authentication requests, where the failure is due to invalid username or password. <i>If all transactions are failing authentication, the problem might be that the shared secret entered into Steel-Belted Radius does not match the shared secret entered on the client device.</i>
Failed on Check List	The total number of requests that were authenticated but failed to meet the Check-List requirements.
Insufficient Resources	The total number of rejects due to a server resource problem.
Proxy Failure	The total number of rejects that had to be issued because Proxy forwarding to another RADIUS server failed.
Rejected by Proxy	The total number of rejects due to receiving a reject response from a Proxy RADIUS target server.
Retries Received	
Transactions Retried	The number of requests for which one or more duplicates was received.
Total Retry Packets	The total number of duplicate packets received.

Accounting Statistics

Accounting statistics provide information such as the number of transaction STARTs and STOPs and the reasons for rejecting attempted transactions. The START and STOP numbers rarely match, as many transactions can be “in progress” at any given time.



Statistics Dialog, Accounting Tab (Windows version)

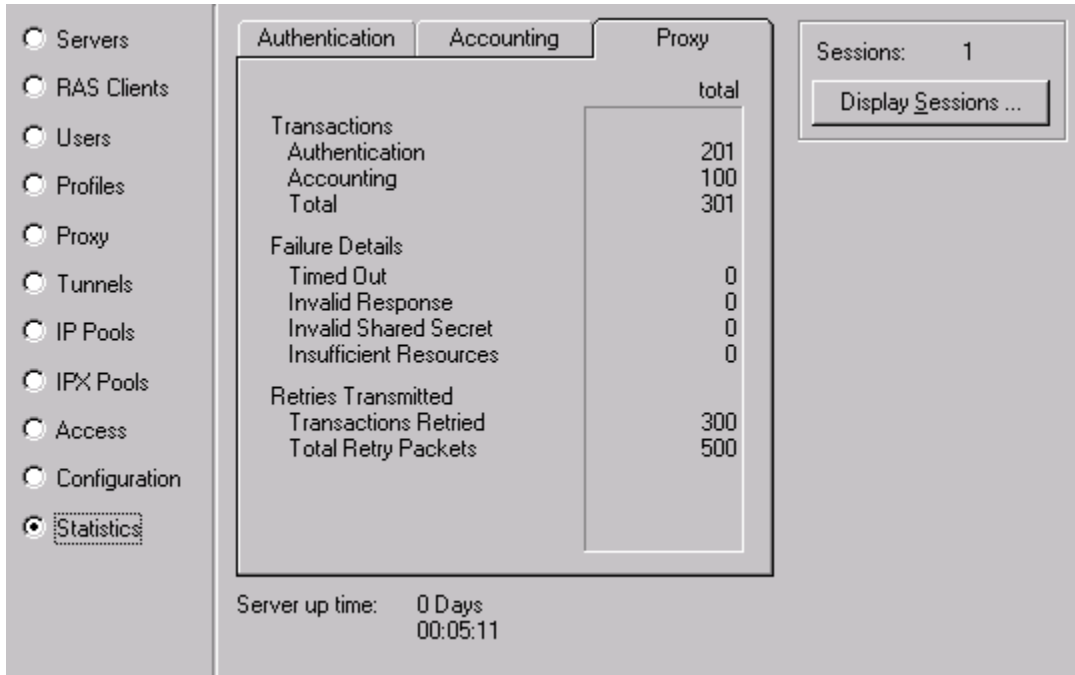
The following table describes the accounting statistics and suggested actions in italics (if appropriate).

Accounting	
Statistic	Meaning
Transactions	
Starts	The total number of transactions in which a dial-in connection was started following a successful authentication.
Stops	The total number of transactions in which a dial-in connection was terminated.
Ons	The total number of Accounting-On messages received, indicating that a RAS client has rebooted.
Offs	The total number of Accounting-Off messages received, indicating that a RAS client has shut down.
Total	The total of the four fields above.
Failure Details	
Invalid Request	The total number of invalid RADIUS requests made. <i>A device is sending incorrectly formed packets to Steel-Belted Radius; either there is a configuration error or the device does not conform to the RADIUS standard.</i>

Accounting Statistic	Meaning
Invalid Client	The total number of requests in which the RAS Client could not be identified. <i>A device might be configured to use Steel-Belted Radius but no RAS Client entry has been created with the name and/or IP address of the client; or the RAS Client entry might be configured with an incorrect name or IP address; or some rogue device is attempting to compromise RADIUS security.</i>
Invalid Shared Secret	The total number of packets for which an incorrect digital signature was received. <i>The shared secret does not match between Steel-Belted Radius and the client device; or some rogue device is attempting to compromise RADIUS security.</i>
Insufficient Resources	The total number of rejects due to a server resource problem.
Proxy Failure	The total number of times that Proxy RADIUS forwarding failed.
Retries Received	
Transactions Retried	The number of requests for which one or more duplicates was received.
Total Retry Packets	The total number of duplicate packets received.

Proxy Statistics

Proxy statistics provide information such as the number of proxy authentication or accounting requests and the reasons for any transaction failures that occur.



Statistics Dialog, Proxy Tab

The following table describes the proxy statistics, with possible interpretations in italics.

Proxy Statistic	Meaning
Transactions	
Authentication	The total number of authentication transactions between the proxy and target RADIUS servers.
Accounting	The total number of accounting transactions between the proxy and target RADIUS servers.
Total	The total of the two fields above.
Failure Details	
Timed Out	The total number of RADIUS transactions that timed out. This means that after all retry attempts were made, the transaction still timed out.

Proxy Statistic	Meaning
Invalid Response	The total number of invalid RADIUS responses received. <i>A target is sending incorrectly formed packets to Steel-Belted Radius; either there is a configuration error or the target RADIUS server does not conform to the RADIUS standard. Or, Steel-Belted Radius did not receive a proxy state echo in the received packet.</i>
Invalid Shared Secret	The total number of packets for which an incorrect digital signature was received. <i>The shared secret does not match between Steel-Belted Radius and the target; or some unauthorized rogue device is attempting to compromise RADIUS security.</i>
Insufficient Resources	The total number of rejects due to a server resource problem.
Retries Transmitted	
Transactions Retried	The number of requests for which one or more retried transmissions was performed.
Total Retry Packets	The total number of duplicate packets received.

Resetting Server Statistics

Although the statistics are automatically reset if you restart the server, you can also request for all statistics to be reset to zero without having to restart the server. How you accomplish this operation depends on your operating system:

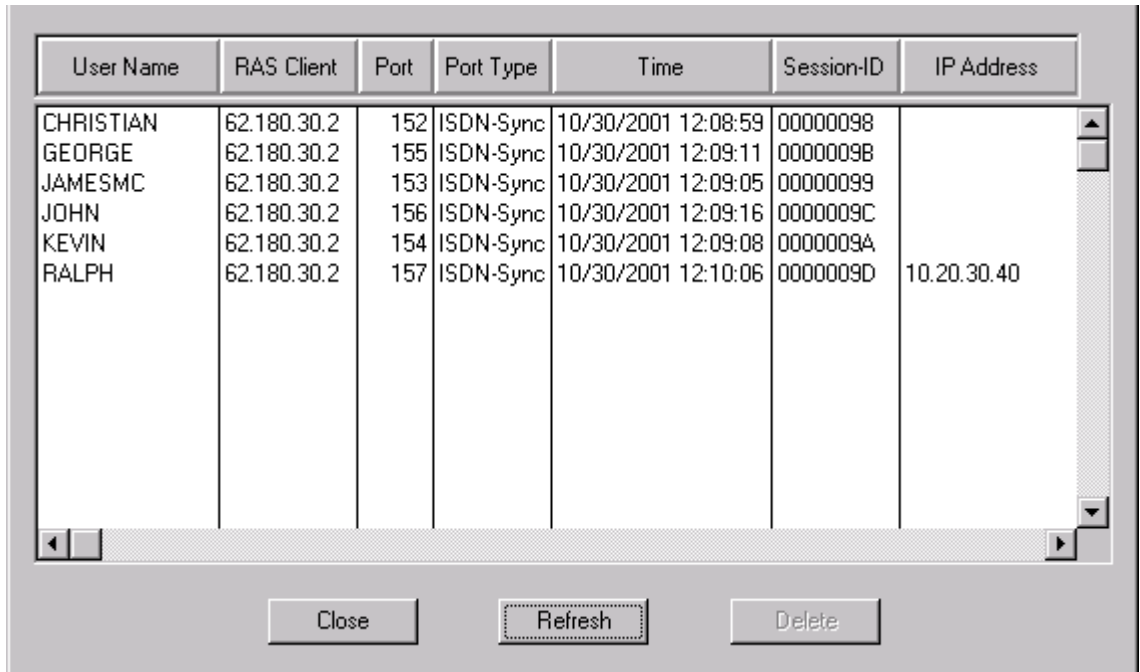
- Under **UNIX**, issue the command (stored in the Radius directory):
kill -USR2 pldServer
where **pldServer** is the process id of your Steel-Belted Radius server.
- Under **Windows**, issue the command (stored in the directory \Radius\Service):
radusr2

Sessions List

Steel-Belted Radius tracks the status of the user connections that it authenticates. You can display the Sessions list (also called the “Current Users display”) by clicking the **Display Sessions** button at the upper right of the Statistics dialog.

The Sessions list is based on RADIUS accounting data. The Sessions list is accurate only if all of your NAS devices are configured to support RADIUS accounting.

Note: Steel-Belted Radius maintains the Current User list on disk. The information is not lost if you unload and reload the server.



User Name	RAS Client	Port	Port Type	Time	Session-ID	IP Address
CHRISTIAN	62.180.30.2	152	ISDN-Sync	10/30/2001 12:08:59	00000098	
GEORGE	62.180.30.2	155	ISDN-Sync	10/30/2001 12:09:11	00000098	
JAMESMC	62.180.30.2	153	ISDN-Sync	10/30/2001 12:09:05	00000099	
JOHN	62.180.30.2	156	ISDN-Sync	10/30/2001 12:09:16	0000009C	
KEVIN	62.180.30.2	154	ISDN-Sync	10/30/2001 12:09:08	0000009A	
RALPH	62.180.30.2	157	ISDN-Sync	10/30/2001 12:10:06	0000009D	10.20.30.40

Close Refresh Delete

Current Users Dialog (Windows version)

For every active dial-in session, the Sessions list displays a line containing the following fields:

- **User Name** shows the name of the authenticated user.
 - If the user is native, the field shows only the username, in the form username.
 - If the user is non-native, the field shows the remote system name as well as the username, in the form \\systemname\username.
 - If the user is associated with a specific tunnel, the field shows the tunnel name as well as the username, in the form \\tunnelname\username.
- **RAS Client** is the NAS's identifier, which is either the name or IP address of the device.
- **Port** is a unique port number on the NAS that has been assigned to the connection. To determine the actual physical port on the NAS, consult the NAS documentation.

- **Port Type** describes how the port is used or configured. Possibilities include Async, Sync, ISDN, and so forth.
- **Time** indicates the date and time at which the connection was started.
- **Session ID** contains the unique key for the session, a number generated by the NAS.
- **IP Address** shows the IP address that was assigned to the user from an IP address pool. (If an IP address was statically assigned, this field is blank.)

Note: For tunnel connections, if Steel-Belted Radius was used to authenticate both the user and the tunnel, then two entries are displayed in the Current Users window: one entry for the authenticated user, and one for the authenticated tunnel.

Modifying the Sessions List Columns

You can modify the order and size of the fields in the Sessions List easily with the mouse.

- To resize any field, move the mouse to the right edge of a field heading. When the resize cursor appears, click and drag the field width as you please.
- To move any field, click on any field heading and drag it left or right to the desired position.

Note: This feature is supported only by Windows.

Sorting the Sessions List (Windows only)

The Sessions List is always maintained in sorted order, based on each column of the report starting from the first column at the left. Thus, to sort the report according to the values in any field, simply drag that field to the leftmost position.

Refreshing the Sessions List

The Sessions List shows a snapshot of the current connected users taken when you first open the dialog. To update the report with fresh information, click **Refresh**.

Deleting Entries from the Sessions List

Normally, the system maintains the information in the Sessions list based on accounting information received from the NAS. However, a user who has logged off might still be identified as active in the Current User list if communications between

the NAS and Steel-Belted Radius fail or if either the NAS or Steel-Belted Radius is taken down for a period of time.

In most cases, Steel-Belted Radius can correct such anomalies itself. For example, if a new user dials in to the same port on the same NAS, Steel-Belted Radius infers that the prior user must have disconnected and removes the entry.

You can also manually correct the Sessions list by highlighting any entry and clicking **Delete**. This removes the user from the list and decrements the user's connection count (if it is being tracked) by one. Any pooled IP or IPX address assigned to the deleted user is returned to the appropriate pool.

Reporting Capabilities

The Report command lets you assemble any of the Steel-Belted Radius database information that is available through the Administrator program's dialogs into a report. For example, you can output to a report all the information you've set up about RAS Clients, Users, Proxies, and the like.

Windows only: The **Report** command outputs the information in Rich Text Format (RTF) to a filename of your choice (normally REPORT.RTF), then opens that report using the word processor of your choice (normally WORDPAD.EXE). You can use the word processor to format the report, save it to an archive, or print it.

Setting Report Options (Windows only)

Before creating your first report, verify that the settings for the output filename and the word processor used to view the report are correct:

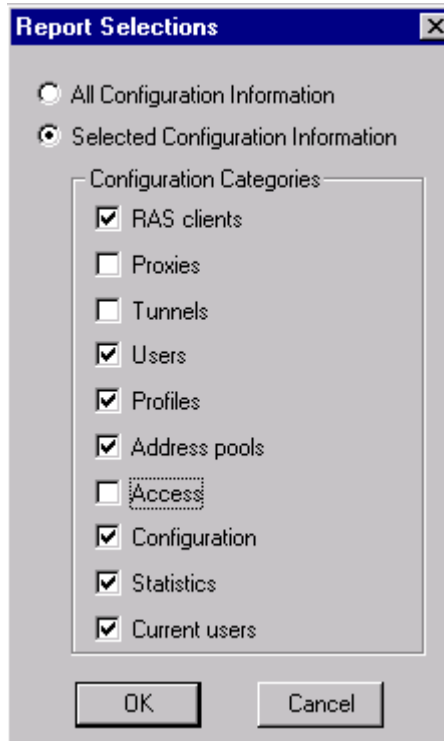
- 1 Run the Administrator program.
- 2 Select the **File Settings** command. The Settings dialog appears.
- 3 Make sure the Report viewer and Report filename settings are to your liking. Be sure that the word processor that you specify as Report viewer is capable of interpreting RTF (rich text format).
- 4 When you are satisfied with the settings, click **OK**.

Creating a Report

To create a report:

- 1 Run the Administrator program.

- 2 Depending on your platform:
 - Under **UNIX**: Click the **Report** button at the bottom of the Steel-Belted Radius Administrator display.
 - Under **Windows**: Select the **File Report** command.
- 3 The Report Selections dialog appears.



Report Selections Dialog (UNIX version)

- 4 To generate a complete report on every aspect of the server, check **All configuration information**.
Otherwise, check **Selected configuration information**, and check the categories of information you'd like to include.
- 5 Click **OK**. Depending on your platform:
 - Under **UNIX**: A new instance of your browser pops up, displaying the resulting report. You can save this report to HTML format if desired.
 - Under **Windows**: The report file is created and appears in your selected word processor.

Windows NT Performance Monitor

The Steel-Belted Radius service has information which can be viewed with the Performance Monitor on Windows NT/2000.

To view a graph of Steel-Belted Radius performance:

- 1 Start `perfmon.exe` on your administrative workstation.

Note: You can also start `perfmon.exe` on a Steel-Belted Radius server machine.

- 2 Select **Edit > Add to Chart**. Select the Steel-Belted Radius service from the list of Objects. If you are running multiple Steel-Belted Radius servers, you can select the correct one by computer name.
- 3 Select the counters that you want to graph. For each counter, choose **Color**, **Scale**, and other display options as desired. Then click **Add**.

Most `perfmon` counters relating to Steel-Belted Radius have self-explanatory names, such as `Acct Failures - Insufficient Resources` or `Acct Failures - Invalid Shared Secret`.

Of special interest is the `Failed Auths - n` counter. There are 16 such counters, where n is a number between 1 and 16. The `Failed Auths - n` `perfmon` counter tracks the total number of failed authentication requests that were encountered for all of the RADIUS clients that you've mapped to collection number n .

To set up the `Failed Auths - n` counter, see “radius.ini [FailedAuthOriginStats] Section (Windows only)” on page 216.

- 4 When you are finished adding counters, click **Done**.

The Performance Monitor window displays a graph of the counters you've selected. The graph updates itself at regular intervals until you close the Performance Monitor window.

- 5 You can start multiple versions of `perfmon.exe` to view more than one Steel-Belted Radius server at one time.

The following `perfmon` counters are available.

perfmon Counter	Meaning
Acct Failures - Insufficient Resources	The number of accounting requests that were discarded because the RADIUS server was unable to obtain sufficient system resources to process the request.
Acct Failures - Invalid Clients	The number of accounting requests that were discarded because the RADIUS client identified in the request was not defined in the RADIUS server database.

perfmom Counter	Meaning
Acct Failures - Invalid Requests	The number of accounting requests that were discarded because the request was malformed or contained invalid attributes.
Acct Failures - Invalid Shared Secret	The number of accounting requests that were discarded because the request contained an invalid digital signature. This is usually due to a mismatch in the shared secrets defined on the RADIUS client and the RADIUS server.
Acct Proxy Failures	The number of forwarded accounting requests for which failures were encountered.
Acct Requests Forwarded	The number of accounting requests that were forwarded to other RADIUS servers.
Acct Requests Retried	The number of unique accounting requests for which retries were received by the RADIUS server.
Acct Requests Retried/sec	The number of accounting requests per second for which one or more retries has been received by the RADIUS server.
Acct Retry Requests	The number of actual accounting request retries received by the RADIUS server.
Acct Retry Requests/sec	The number of accounting request retries per second received by the RADIUS server.
Acct Service Time	The number of seconds that elapsed from the time the last completed accounting request was received to the time the RADIUS server sent a response. Responses generated by proxies are not reflected in this statistic.
Acct Starts	The number of accounting start requests received by the RADIUS server. An accounting start signifies the granting of a connection to an end-user by the remote access server.
Acct Starts/sec	The number of accounting start requests received by the RADIUS server per second. An accounting start signifies the granting of a connection to an end-user by the remote access server.
Acct Stops	The number of accounting stop requests received by the RADIUS server. An accounting stop signifies that an end-user has disconnected from the remote access server.
Acct Stops/sec	The number of accounting stop requests received by the RADIUS server per second. An accounting stop signifies that an end-user has disconnected from the remote access server.
Auth Failure - Authentication Failures	Number of unique authentication requests to which the RADIUS server replied with a reject because no user specified in the database possessed a matching password. A mismatch in shared secrets would also cause this counter to be incremented.
Auth Failure - Checklist Mismatches	Number of unique authentication requests to which the RADIUS server replied with a reject because the request did not include required checklist information.

perfmon Counter	Meaning
Auth Failure - Insufficient Resources	Number of unique authentication requests to which the RADIUS server replied with a reject because the RADIUS server ran into a system resource limitation.
Auth Failure - Invalid Clients	Number of unique authentication requests to which the RADIUS server replied with a reject because the request was from a RADIUS client not identified in the RADIUS server's database.
Auth Failure - Invalid Requests	Number of unique authentication requests to which the RADIUS server replied with a reject because the request was malformed or contained invalid attributes.
Auth Proxy Failures	The number of forwarded authentication requests for which failures were encountered.
Auth Proxy Rejects	The number of forwarded authentication requests for which rejects were received from the target RADIUS server.
Auth Requests	The number of unique authentication requests that the RADIUS server has received.
Auth Requests Forwarded	The number of authentication requests that were forwarded to another RADIUS server.
Auth Requests Retried	The number of unique authentication requests for which retries were received by the RADIUS server.
Auth Requests Retried/sec	The number of authentication requests per second for which one or more retries has been received by the RADIUS server.
Auth Requests/sec	The number of unique authentication requests that the RADIUS server has received per second.
Auth Retry Requests	The number of actual authentication request retries received by the RADIUS server.
Auth Retry Requests/sec	The number of authentication request retries per second received by the RADIUS server.
Auth Service Time	The number of seconds that elapsed from the time the last completed authentication request was received to the time the RADIUS server sent an Accept response. Accept responses generated for tunnel requests or by proxies are not reflected in this statistic.
Auth SQL Disconnects	The number of times an existing connection to a SQL authentication database failed.
Auth SQL Failures	The number of times an attempt to connect to a SQL authentication database failed.
Auth SQL Records Not Found	The number of times no record was found in a SQL authentication database for the specified username.
Auth SQL Timeouts	The number of times a timeout occurred attempting to execute a SQL authentication request.
Auth Successes	The number of unique authentication requests to which the RADIUS server replied with an Accept.

perfmom Counter	Meaning
Auth Successes/sec	The number of unique authentication requests to which the RADIUS server replied with an accept per second.
Concurrency Auth Failures	The number of times an authentication request was forwarded to the concurrency server and the concurrency server returned a reject for reasons other than users being over their port limits.
Concurrency Auth Service Time	The number of seconds elapsed from the last time an authentication request was sent to the concurrency server and a response was received.
Concurrency Auth Timeouts	The number of times an authentication request was forwarded to the concurrency server and no response was received within the configured time for the proxy entry.
Concurrency Over Port Limit	The number of times an authentication request was forwarded to the concurrency server and the concurrency server returned a reject because users were over their port limits.
Failed Auths - 1	The number of failed authentication requests that were encountered for clients categorized in collection number 1. To set up this counter, see "radius.ini [FailedAuthOriginStats] Section (Windows only)" on page 216.
Failed Auths - 2	The number of failed authentication requests that were encountered for clients categorized in collection number 2.
Failed Auths - 3	The number of failed authentication requests that were encountered for clients categorized in collection number 3.
Failed Auths - 4	The number of failed authentication requests that were encountered for clients categorized in collection number 4.
Failed Auths - 5	The number of failed authentication requests that were encountered for clients categorized in collection number 5.
Failed Auths - 6	The number of failed authentication requests that were encountered for clients categorized in collection number 6.
Failed Auths - 7	The number of failed authentication requests that were encountered for clients categorized in collection number 7.
Failed Auths - 8	The number of failed authentication requests that were encountered for clients categorized in collection number 8.
Failed Auths - 9	The number of failed authentication requests that were encountered for clients categorized in collection number 9.
Failed Auths - 10	The number of failed authentication requests that were encountered for clients categorized in collection number 10.
Failed Auths - 11	The number of failed authentication requests that were encountered for clients categorized in collection number 11.
Failed Auths - 12	The number of failed authentication requests that were encountered for clients categorized in collection number 12.
Failed Auths - 13	The number of failed authentication requests that were encountered for clients categorized in collection number 13.

perfmon Counter	Meaning
Failed Auths - 14	The number of failed authentication requests that were encountered for clients categorized in collection number 14.
Failed Auths - 15	The number of failed authentication requests that were encountered for clients categorized in collection number 15.
Failed Auths - 16	The number of failed authentication requests that were encountered for clients categorized in collection number 16.
Forwarded Requests Retried	The number of unique forwarded accounting and authentication requests for which retries were transmitted by the RADIUS server.
Forwarded Requests Retried/sec	The number of unique forwarded accounting and authentication requests for which retries were transmitted per second by the RADIUS server.
Forwarded Retry Requests	The number of actual retransmissions of forwarded accounting and authentication request performed by the RADIUS server.
Forwarded Retry Requests/sec	The number of actual retransmissions of forwarded accounting and authentication request per second performed by the RADIUS server.
Proxy Failures - Insufficient Resources	The number of authentication and accounting requests that were forwarded to other RADIUS servers for which the RADIUS server was unable to obtain sufficient system resources to process the request.
Proxy Failures - Invalid Response	The number of authentication and accounting requests that were forwarded to other RADIUS servers for which malformed or invalid responses were received.
Proxy Failures - Invalid Shared Secret	The number of authentication and accounting requests that were forwarded to other RADIUS servers for which responses were discarded because the response contained an invalid digital signature. This is usually due to a mismatch in the shared secrets defined on the RADIUS client and RADIUS server.
Proxy Failures - Time Out	The number of authentication and accounting requests that were forwarded to other RADIUS servers for which no response was received after the specified number of retries.
Seconds since started	Number of seconds Steel-Belted Radius has been running.
Sessions Online	The number of sessions currently active in the RADIUS server's Sessions list.
Static Acct Service Time	The number of seconds elapsed from the last time an accounting request was sent to the static accounting proxy server and a response was received.
Total Acct Failures	The total number of unique accounting requests to which the RADIUS server did not reply. Reasons for the failures are identified in other statistics.

perfmom Counter	Meaning
Total Acct Failures/sec	The total number of unique accounting requests to which the Radius server did not reply per second because of an error. Reasons for the failures are identified in other statistics.
Total Acct Offs	The number of accounting off requests received by the RADIUS server. An accounting off signifies that the accounting support in the RADIUS client has been disabled. This request is most often issued when a RADIUS client is being shut down.
Total Acct Offs/sec	The number of accounting off requests received by the RADIUS server per second.
Total Acct Ons	The number of accounting on requests received by the RADIUS server. An accounting on signifies that the accounting support in the RADIUS client has been enabled. This request is most often issued when a RADIUS client is powered on.
Total Acct Ons/sec	The number of accounting on requests received by the RADIUS server per second.
Total Auth Challenges	The number of authentication requests that resulted in a RADIUS challenge response.
Total Auth Failures	The total number of unique authentication requests to which the RADIUS server replied with a reject. Reasons for the failures are identified in other statistics.
Total Auth Failures/sec	The total number of unique authentication requests to which the RADIUS server replied with a reject, per second. Reasons for the failures are identified in other statistics.
Total Forwarded Request Failures	The total number of forwarded authentication and accounting requests that encountered failures.
Total Forwarded Request Failures/sec	The total number of forwarded authentication and accounting requests that encountered failures, per second.
Total Forwarded Requests	The total number of authentication and accounting requests that were forwarded to other RADIUS servers.
Total Forwarded Requests/sec	The total number of authentication and accounting requests per second that were forwarded to other RADIUS servers.
Users Online	The number of unique user names represented in the RADIUS server's Sessions List.

Windows NT Events

Steel-Belted Radius generates a variety of Windows NT events. Regardless of severity, each event is attributed to one of the following three Windows NT services:

the “core” Steel-Belted Radius service, the authentication service, or the accounting service. Service identifiers are as follows.

ID	Symbolic Name	Text
1	RADCAT_CORE	Core
2	RADCAT_AUTH	Authentication
3	RADCAT_ACCT	Accounting

The following three topics group Steel-Belted Radius events according to their severity: Informational, Warning, and Error.

Informational Events

The following events are for informational purposes only. They do not require intervention by an operator.

Note: Some informational events serve to “clear” a previous warning event.

ID	Informational Event	Meaning
100	The Steel-Belted Radius service was started.	—
101	The Steel-Belted Radius service was stopped.	—
102	Count of available threads has risen to acceptable threshold of <i>nnnn</i> .	The low thread available condition has subsided. You can configure the value <i>nnnn</i> using the [Thresholds] section of the events.ini configuration file.
103	Amount of free file system space has risen to acceptable threshold. Free byte count is <i>nnnnnnnn</i> .	The low file system space condition has subsided. You can configure the value <i>nnnnnnnn</i> using the [Thresholds] section of events.ini.
104	Steel-Belted Radius has reconnected to the Concurrency Server after a ConcurrencyFailure.	—
105	Steel-Belted Radius has reconnected to the SQL database after a SQLConnectFail.	—
106	Steel-Belted Radius has reconnected to the LDAP database after an LDAPConnectFail.	—

ID	Informational Event	Meaning
107	A user's account has been locked due to excessive authentication attempts within a defined period of time.	—
108	A user account, previously locked due to an excessive amount of rejected authentication attempts, becomes unlocked.	—
109	The target server for proxy spooling reconnected.	—

Warning Events

Warning events can require intervention, as indicated in the following table.

The text `This event represents nnnn failures` reflects a setting in the [EventDilutions] section of the `events.ini` file. You can set this value to a higher number so that the event is reported less frequently.

ID	Warning Event	Meaning
5001	Count of available threads has dropped to minimum threshold of <i>nnnn</i> .	A low thread count available condition has been detected. This event can be issued in the authentication or accounting category to indicate a shortage of authentication or accounting threads. You can configure the value <i>nnnn</i> using the [Thresholds] section of the <code>events.ini</code> configuration file.
5002	Concurrency server returned failure indication. This event represents <i>nnnn</i> failures.	A reject was returned from the concurrency server in response to a proxied authentication request. The reject was for a reason other than exceeded port limit. You can configure the value <i>nnnn</i> using the [EventDilutions] section of <code>events.ini</code> .
5003	Timed out in proxy attempt to concurrency server. This event represents <i>nnnn</i> requests timing out.	A timeout was encountered when proxy-forwarding an authentication request to the concurrency server. You can configure the value <i>nnnn</i> using the [EventDilutions] section of <code>events.ini</code> .
5004	Local failure encountered in attempt to proxy to concurrency server. This event represents <i>nnnn</i> requests timing out.	A local processing failure was encountered when trying to proxy-forward an authentication request to the concurrency server. You can configure the value <i>nnnn</i> using the [EventDilutions] section of <code>events.ini</code> .

ID	Warning Event	Meaning
5005	Timed out in static accounting proxy attempts. This event represents <i>nnnn</i> failures.	A timeout was encountered when proxy-forwarding an accounting request to the concurrency server. You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
5006	Local failure encountered in attempt to proxy for static accounting. This event represents <i>nnnn</i> requests timing out.	A local processing failure was encountered when trying to proxy-forward an accounting request to the concurrency server. You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
5007	Amount of free file system space has dropped below minimum threshold. Free byte count is <i>nnnnnnnn</i> .	A low available file system space condition has been detected. You can configure the value <i>nnnnnnnn</i> using the [Thresholds] section of events.ini.
5008	<i>nnnn</i> attempts to connect to SQL server failed.	You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
5009	<i>nnnn</i> disconnects from SQL server due to error.	You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
5010	<i>nnnn</i> timeouts on SQL requests.	You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
5011	Access to accounting server database has timed out. This event represents <i>nnnn</i> timeouts.	You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
5012	Access to accounting server database has failed. This event represents <i>nnnn</i> failures.	You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
5013	Verification Server has timed out. This event represents <i>nnnn</i> Verification Server timeouts.	You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
5014	Verification Server requests have failed. This event represents <i>nnnn</i> Verification Server failures.	You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
5015	The connection to an LDAP server has failed.	—
5016	Communication with an LDAP server has failed. This event represents <i>nnnn</i> connection failures.	You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.

ID	Warning Event	Meaning
5017	The LDAP server has disconnected. This event represents <i>nnnn</i> connection failures.	You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
5018	A request to the LDAP server has timed out. This event represents <i>nnnn</i> request timeouts.	You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
5019	The LDAP server has disconnected	—
5020	A request to the LDAP server has timed out.	—
5021	The target server of proxy spooling fails to respond (non-dilutable).	—
5022	The target server of proxy spooling fails to respond (dilutable).	—

Error Events

Error events usually require some form of intervention by the operator.

Most Steel-Belted Radius error events are generated at startup, as the service initializes each of its components. If any component fails at startup, the “service start” operation is aborted and the system generates an error event. The text of the error identifies what Steel-Belted Radius was doing when it failed.

In some cases, the operator can take direct action in response to an error event. For example, if the system is unable to open a log file, the system disk might be full, leaving no room to create additional files.

If a solution does not emerge immediately, the event text identifies the problem area in the software. You should escalate the problem to your next level of support. When you do, be sure to indicate the ID, name, and text of the event.

ID	Error Event
10000	StartServiceCtrlDispatcher failed with error <i>nnnn</i> .
10001	SetServiceStatus failed with error <i>nnnn</i> .
10002	Invalid private directory 'directory' specified.
10003	Unable to create thread.
10004	Unable to create mutex.

ID	Error Event
10005	Unable to initialize signal handling.
10006	Unable to configure event processing.
10007	Unable to create or open log file.
10008	Unable to initialize LDAP administration interface.
10009	Unable to initialize RPC administration interface.
10010	Unable to initialize base IP interface.
10011	Unable to initialize current user list processing.
10012	Unable to initialize challenge continuation cache.
10013	Unable to initialize RAS activity monitor.
10014	Unable to initialize dictionary processing.
10015	Unable to process vendor.ini file.
10016	Unable to initialize Btrieve Raima database.
10018	Unable to initialize admin user rights component.
10019	Unable to open Btrieve Raima database.
10020	Unable to initialize tunnel DNIS lookup component.
10021	Unable to initialize configuration caching component.
10022	Unable to initialize database caching component.
10023	Unable to initialize license processing.
10024	Unable to initialize NDS trustee processing.
10025	Unable to initialize NetWare host lookup processing.
10026	Unable to initialize IP/IPX pool resource management.
10027	Unable to initialize user login count tracking.
10028	Unable to create persistent store for Sessions list.
10029	Unable to initialize persistent store for Sessions list.
10030	Unable to initialize performance monitor interface component.
10031	Unable to initialize admin locking component.
10032	Unable to initialize plug-in support component.
10033	Unable to initialize duplicate request cache.
10034	Unable to initialize name mangling support.
10035	Unable to initialize name stripping support.
10036	Error <i>nnnn</i> returned from call to GetDiskFreeSpaceEx. File system space checking disabled.
10037	Unable to initialize name validation support. Service start aborted.
10038	Unable to initialize system resource checking. Service start aborted.
10039	Unable to initialize statistic collection. Service start aborted.
10040	Attempt to connect to SQL server <i>xxxxxxxxxx</i> failed.

ID	Error Event
10041	Disconnect from SQL server <i>xxxxxxxxxx</i> due to error.
10042	Timeout on SQL request.
10043	Unable to allocate reserved memory specified by ReserveMemoryKB. You can set the ReserveMemoryKB value in the [Thresholds] section of the events.ini configuration file.
10044	Memory allocation failure encountered. Reserved memory released as last resort.
10045	<i>nnnn</i> memory allocation failures have occurred. You can configure the value <i>nnnn</i> using the [EventDilutions] section of events.ini.
10046	The connection to the Accounting Server has failed.
10047	The connection to the Verification Server has failed.
10050	The initialization of common IP services at server startup has failed.

Server Configuration

6

- Server Configuration Files
- access.ini File
- account.ini File
- admin.ini File
- authlog.ini File
- blacklist.ini File
- bounce.ini File (Windows only)
- classmap.ini File
- dhcp.ini File
- pool.dhc Files
- events.ini File
- filter.ini File
- lockout.ini File
- radius.ini File
- redirect.ini File
- spi.ini File
- tacplus.ini File
- update.ini File
- vendor.ini File
- Dictionary Files
- services File
- Attribute Value Pools (*.rr files)
- Auto-Restart Files (UNIX only)

Server Configuration Files

Server configuration files control the behavior of the Steel-Belted Radius server. While you should learn about these files in order to customize the operation of the server for advanced operation, the default settings allow you to run a generic configuration of the server immediately after installation without requiring you to alter these files.

Server configuration files include:

- Initialization files, which enable, disable, and configure various features of the server. These files are loaded at startup time, and reside in the Steel-Belted Radius server directory:

- access.ini
 - account.ini
 - admin.ini
 - blacklist.ini
 - bounce.ini (Windows only)
 - classmap.ini
 - dhcp.ini
 - eap.ini
 - events.ini
 - filter.ini
 - lockout.ini
 - radius.ini
 - spi.ini
 - tacplus.ini
 - vendor.ini

- Dictionary files, which specify RADIUS attributes. Like initialization files, these files are loaded at startup time, and reside in the Steel-Belted Radius server directory:

- *.dct
 - *.dci
 - dictiona.dcm

- Automatic EAP Helper configuration files, which specify options for automatic EAP helper methods. These files are loaded at startup time and reside in the Steel-Belted Radius server directory:

- *.eap

- The services file, which assigns default UDP ports for RADIUS communications to and from the Steel-Belted Radius server. The location of the file is:

- (Under **UNIX**) /etc/

- (Under **Windows**) C:\winnt\system32\drivers\etc\
- (**UNIX** Only) The S90radius and radiusd scripts in the /etc/rc2.d directory. These scripts enable and configure the auto-restart module for the Steel-Belted Radius server.

Other files control the server's interactions with "outside parties," including:

- External databases, for authentication and accounting respectively:
 - *.aut
 - *.acc
- Realms, a flexible, organizing concept that can support Proxy RADIUS or other types of authentication and accounting services:
 - proxy.ini
 - RealmName.pro
 - RealmName.dir

The pound ('#') character comments out a line, as long as the '#' is the first non-space character in the line. The semicolon (;) can be used as a comment character in the same way.

access.ini File

The access.ini initialization file maps operating system user or group account names to levels of administrative privilege. The correct syntax for the [Users] and [Groups] sections of this file is as follows:

```
[Users]
UserName = AccessLevel
UserName = AccessLevel
.
.
.
[Groups]
GroupName = AccessLevel
GroupName = AccessLevel
.
.
```

Field	Meaning
UserName GroupName	Each <i>UserName</i> or <i>GroupName</i> is the name of an authorized administrator account on the server. You must list user accounts in the [Users] section, group accounts in the [Groups] section. You should list groups in priority order; rights are granted based on the first group found of which the user is a member.
AccessLevel	The <i>AccessLevel</i> in each access.ini entry is the access level that you want to assign to that account. Each <i>AccessLevel</i> string must exactly match the name of an [AccessLevel] section in admin.ini. You can define as many or as few [AccessLevel] sections as you want. Once an [AccessLevel] section is defined in admin.ini, you can use access.ini to assign this AccessLevel to various user and group accounts.

See “admin.ini File” on page 184.

The [Settings] section of access.ini contains overall configuration parameters; do not edit this section.

Note: The access.ini.sample file in the server directory can be used as a template for your own access.ini file.

There is a default access level called SuperAdmin which grants read/write access to all types of administrative data. This access level is always defined, and can be assigned to a user or group account in access.ini without appearing in admin.ini.

Note: (UNIX only) The default Steel-Belted Radius administrative account (admin) has SuperAdmin rights.

account.ini File

The account.ini initialization file contains information that controls how RADIUS accounting attributes are logged by Steel-Belted Radius.

account.ini [Alias/name] Sections

The [Alias/name] sections of account.ini are used to associate attributes of different names, but identical meaning. For example, one NAS vendor might call an attribute Acct-Octet-Pkt and another might call it Acct-Oct-Packets, yet the two attributes mean the same thing.

Each [Alias/name] section permits you to map one RADIUS accounting attribute that is already being logged by Steel-Belted Radius to any number of other

attributes. You can provide as many [Alias/*name*] sections as you want, using the following syntax for each section:

```
[Alias/name]
VendorSpecificAttribute=
VendorSpecificAttribute=
.
.
.
```

Field	Meaning
name	The preferred attribute name. The name attribute must be one that you are currently logging to a column in the Steel-Belted Radius accounting log file (.ACT). Therefore, it must be listed in the [Attributes] section of account.ini.
VendorSpecificAttribute	Each entry is given on one line. An equal sign ('=') must immediately follow each VSA name, without any intervening space. Improperly formatted entries are considered invalid and are ignored.

Each *VendorSpecificAttribute* in the list is logged to the *name* column in the accounting log file. Because you're listing these attributes in an [Alias/*name*] section, you should verify they are not listed in the [Attributes] section, or they will be logged to their own columns as well. as the *name* column.

All of the attribute names that you reference in an [Alias/*name*] section must be defined in a dictionary file that is already installed on the Steel-Belted Radius server. This includes *name* and each *VendorSpecificAttribute* entry.

In the following example, the standard RADIUS attribute *Acct-Octet-Packets* is mapped to the vendor-specific attributes *Acct-Octet-Pkt* and *Acct-Oct-Packets*. Values encountered for all three attributes are logged in the *Acct-Octet-Packets* column in the accounting log file:

```
[Alias/Acct-Octet-Packets]
Acct-Octet-Pkt=
Acct-Oct-Packets=
```

account.ini [Attributes] Section

The [Attributes] section of account.ini lists all the attributes logged in the accounting log file. When you first install Steel-Belted Radius, the account.ini file is set up so that all standard RADIUS attributes and all supported vendors' accounting attributes are listed.

You can configure what is logged to the accounting log file by rearranging the order of attributes in the [Attributes] section. You can delete or comment out any attributes that are not of interest to your billing system or which do not apply to the equipment

that you are using. This lets you design the content and column order of any spreadsheets that you plan to create based upon the accounting log file.

The syntax is as follows:

```
[Attributes]
AttributeName=
AttributeName=
.
.
.
```

For example:

```
[Attributes]
User-Name=
NAS-Port=
Framed-IP-Address=
Acct-Status-Type=
Acct-Delay-Time=
Acct-Session-Id=
```

The [Attributes] section lists one *AttributeName* on each line. You must ensure that an equal sign (=) immediately follows each *AttributeName*, with no spaces in between. Improperly formatted entries are considered invalid and are ignored.

Each *AttributeName* in the [Attributes] section must be defined in a dictionary file (.dct) on the Steel-Belted Radius server. This dictionary can be standard RADIUS or vendor-specific.

Note: The first six attributes in each log file entry (Date, Time, RAS-Client, Record-Type, Full-Name, and Auth-Type) are always enabled, and cannot be re-ordered or deleted. Therefore, these attributes do not appear in the account.ini file [Attributes] section.

account.ini [Configuration] Section

account.ini [Configuration]	
field	Meaning
LogDir	If this setting is present, it overrides the default system location (the private directory). You may need to add this section and field if it does not exist in your account.ini file. <i>NOTE: With directed realms, you can still maintain multiple accounting log locations.</i>

account.ini [Settings] Section

Steel-Belted Radius writes all accounting data to the current accounting log file (.ACT) until that log file is closed; then upon closing the file Steel-Belted Radius opens a new one and begins writing accounting data to it.

You can configure how often this “rollover” of the accounting log file occurs.

The naming conventions of the accounting log files support the fact that there can be more than one file per day. The formats are as follows. In the examples below, *y*=year digit, *m*=month digit, *d*=day digit, and *h*=hour digit. The extra sequence number *_nnnnn* starts at *_00000* each day.

File Generation Method	File Naming Convention
Default (24 hours)	<i>yyyymmdd</i> .ACT
Non-24-hour rollover	<i>yyyymmdd_hhmm</i> .ACT
Rollover due to size	<i>yyyymmdd_nnnnn</i> .ACT
Rollover due to size or startup when non-24-hour time in effect	<i>yyyymmdd_hhmm_nnnnn</i> .ACT

The following fields have been provided for use in the [Settings] section of account.ini. These fields control which entries are written to the accounting log file, and ensure the compatibility of these entries with a variety of database systems. The following “rollover” fields can be present in the [Settings] section.

account.ini [Settings] field	Meaning
BufferSize	The size of the buffer used in the accounting logging process, in bytes. The default is 32768.
Carryover	If set to 1 (the default), each time a new accounting log file is created, a summary of the Sessions List is written to the file. If 0, the list is not written.
Enable	If set to 1 (the default), the accounting log feature is enabled. If 0, no .ACT files are created on this server. Accounting servers should have Enable set to 1; for efficiency, non-accounting servers should have Enable set to 0.
LineSize	The maximum size of a single accounting log line. The default is 4096. The allowable range is 1024 to 32768.
MaxSize	The maximum size of an accounting log file, in bytes. Once the accounting log file reaches this limit, the log file is closed and a new file started. A value of 0 (the default) means unlimited size.

account.ini [Settings] field	Meaning
QuoteBinary	If set to 1 (the default), binary values written to the accounting log file are enclosed in quotes; if 0, quotes are not used. You should set this value according to the format expected by the accounting application that will eventually receive the entries.
QuoteInteger	If set to 1 (the default), integer values written to the accounting log file are enclosed in quotes. If 0, quotes are not used. You should set this value according to the format expected by the accounting application that will eventually receive the entries.
QuoteIPAddress	If set to 1 (the default), IP addresses written to the accounting log file are enclosed in quotes; if 0, quotes are not used. You should set this value according to the format expected by the accounting application that will eventually receive the entries.
QuoteText	If set to 1 (the default), text strings written to the accounting log file are enclosed in quotes; if 0, quotes are not used. You should set this value according to the format expected by the accounting application that will eventually receive the entries.
QuoteTime	If set to 1 (the default), time and date values written to the accounting log file are enclosed in quotes; if 0, quotes are not used. You should set this value according to the format expected by the accounting application that will eventually receive the entries.
Rollover	How often the current accounting log file is closed and a new file opened (a rollover), up to one rollover per minute. Non-zero values indicate the number of minutes until the next rollover. A value of 0 (the default) causes a rollover once every 24 hours, at midnight local time.
RolloverOnStartup	If set to 1, each time Steel-Belted Radius is started up, it closes the current accounting log file and open a new one. A sequence number <i>_nnnnn</i> is appended to the log file name, just as when MaxSize is reached. If 0 (the default), each time Steel-Belted Radius is started up, it appends entries to the previously open accounting log file.
Titles	If set to 1 (the default), each time a new accounting log file is created, the title line (containing column headings) is written to the file. If 0, the line is not written.
UTC	If set to 1, time and date values are provided according to universal time coordinates (UTC, formerly known as Greenwich mean time or GMT). If 0 (the default), time and date values reflect local time.

account.ini [TypeNames] Section

Each entry in the [TypeNames] section of account.ini maps a possible value of the Acct-Status-Type attribute to a string. The syntax is as follows:

```
[TypeNames]
TypeID = TypeName
TypeID = TypeName
.
.
.
```

where the fields have meaning as follows:

account.ini

[TypeNames] Field

TypeID	Each <i>TypeID</i> is a numeric value that corresponds to a possible value of the Acct-Status-Type attribute. This attribute appears in every incoming RADIUS accounting packet to identify the types of data it is likely to contain.
TypeName	Each <i>TypeName</i> value is a string. This string is written to the accounting log to identify the type of packet.

The standard Acct-Status-Type values 1, 2, 3, 7, and 8 are already listed in the [TypeNames] section of account.ini as follows:

```
[TypeNames]
1=Start
2=Stop
3=Interim
7=On
8=Off
```

You can edit the [TypeNames] section to add vendor-specific packet types to this list, which makes your accounting log files easier to read and use. For example:

```
[TypeNames]
1=Start
2=Stop
3=Interim
7=On
8=Off
639=AscendType
28=3ComType
```

If no string is given for a particular Acct-Status-Type, Steel-Belted Radius uses the numeric value of the incoming Acct-Status-Type attribute, formatted as a string.

admin.ini File

The admin.ini initialization file maps administrative access levels to sets of access rights. The access levels can then be assigned to administrative accounts by editing the access.ini file.

The full syntax for each access level defined in admin.ini is as follows:

```
[AccessLevel]
Server = Rights
RAS-Clients = Rights
Users = Rights
Profiles = Rights
Proxy = Rights
Tunnels = Rights
IP-Pools = Rights
IPX-Pools = Rights
Configuration = Rights
Access = Rights
Statistics = Rights
CurrentUsers = Rights
Report = Rights
ImportExport = Rights
License = Rights
```

Field	Meaning
Rights	A file privilege: 'r' for read-only access; 'w' for write-only access; or 'rw' for read/write access.
AccessLevel	The name of the access level. The choice of AccessLevel name is arbitrary; you can use any convention that makes sense to you.

Access rights are defined according to the categories of administrative data that an account is allowed to read and/or write. These data categories correspond to Steel-Belted Radius Administrator dialogs and to the objects directly under o=radius in the LDAP configuration schema.

Note: The categories are the same in both cases.

Each [AccessLevel] section must use the keywords listed below. It is acceptable to omit any of these keywords; if a keyword is omitted, access to that data category is specifically denied for all accounts that are assigned that access level.

Note: Misspelled keywords are considered omitted.

The meaning of each [AccessLevel] keyword is as follows.

admin.ini [AccessLevel] Keyword	Meaning
Access	<p>Read or write administrative access data. This is access data controlled by the Administrator program Access dialog. It is distinct from the access data that is controlled by the access.ini and admin.ini files.</p> <p>Each time administrative access is attempted, entries in the Access dialog are consulted first before the .ini files are consulted. Therefore, if an administrative account is listed in the Access dialog, it is granted full access to the administrative database, regardless of how the access.ini and admin.ini files are configured.</p>
Configuration	Read or write miscellaneous information found in the Administrator program Configuration dialog.
CurrentUsers	Read access allows the user to displays the Sessions List. Write access allows the user to delete entries from the Sessions List.
ImportExport	<p>This field controls whether the Import and/or Export menu items are enabled in the Administrator program user interface. Read access allows file export. Write access allows file import.</p> <p><i>NOTE: Import and Export are also subject to the particular rights the user has to each type of item (Users, Tunnels, and so forth).</i></p> <p>Data categories without read access are disabled. If a user tries to export all categories of data, certain categories are omitted from the export operation, if the user does not have read access to that category. The same is true for import.</p>
IP-Pools	Read or write IP address pool data.
IPX-Pools	Read or write IPX address pool data.
License	Write access allows the user to add a new license. Read access is not applicable.
Profiles	Read or write profile data.
Proxy	Read or write Proxy data.
RAS-Clients	Read or write RAS Client data.
Report	Read access enables reports to be generated. Checkbox items for data categories without read access are disabled. If a user tries to collect all categories of data, access to certain categories returns a "Not Authorized" error code if the user does not have read access to that category. Write access is not applicable.
Statistics	Allows the user read access to Authentication, Accounting, and Proxy statistics generated by the server. Write access is not applicable.
Tunnels	Read or write Tunnel data.

admin.ini	
[AccessLevel]	
Keyword	Meaning
Users	Read or write Users data.

Note: The effect of access levels on the LDAP Configuration Interface is that write-only rights allow Add and Modify operations, but not Search operations, on that category of data.

Note: The `admin.ini.sample` file in the server directory can be used as a starting point for your own `admin.ini` file.

authlog.ini File

The `authlog.ini` initialization file contains information that controls how RADIUS authentication request attributes are logged by Steel-Belted Radius.

authlog.ini [Alias/name] Sections

The `[Alias/name]` sections of `authlog.ini` are used to associate attributes of different names, but identical meaning. For example, one NAS vendor might call an attribute `Auth-Octet-Pkt` and another might call it `Auth-Oct-Packets`, yet the two attributes mean the same thing.

Each `[Alias/name]` section permits you to map one RADIUS authentication request attribute that is already being logged by Steel-Belted Radius to any number of other attributes. You can provide as many `[Alias/name]` sections as you want, using the following syntax for each section:

```
[Alias/name]
VendorSpecificAttribute=
VendorSpecificAttribute=
.
.
.
```

Field	Meaning
name	The preferred attribute name. The name attribute must be one that you are currently logging to a column in the Steel-Belted Radius authentication request log file (.authlog). Therefore, it must be listed in the [Attributes] section of authlog.ini.
VendorSpecificAttribute	Each entry is given on one line. An equal sign (=) must immediately follow each VSA name, without any intervening space. Improperly formatted entries are considered invalid and are ignored.

Each *VendorSpecificAttribute* in the list is logged to the *name* column in the authentication request log file. Because you're listing these attributes in an [Alias/*name*] section, you should make sure they are not listed in the [Attributes] section or they will be logged to their own columns as well as to the *name* column.

All of the attribute names that you reference in an [Alias/*name*] section must be defined in a dictionary file that is already installed on the Steel-Belted Radius server. This includes *name* and each *VendorSpecificAttribute* entry.

In the following example, the standard RADIUS attribute `Auth-Octet-Packets` is mapped to the vendor-specific attributes `Auth-Octet-Pkt` and `Auth-Oct-Packets`. Values encountered for all three attributes are logged in the `Auth-Octet-Packets` column in the authentication request log file:

```
[Alias/Auth-Octet-Packets]
Auth-Octet-Pkt=
Auth-Oct-Packets=
```

authlog.ini [Attributes] Section

The [Attributes] section of authlog.ini lists all the attributes logged in the authentication request log file. When you install Steel-Belted Radius, the authlog.ini file is set up so that all standard RADIUS attributes and all supported vendors' authentication attributes are listed.

You can configure what is logged to the authentication request log file by rearranging the order of attributes in the [Attributes] section. You can delete or comment out attributes you do not want or that do not apply to your equipment. This lets you design the content and column order of any spreadsheets that you plan to create based upon the authentication request log file.

The syntax of the [Attributes] section is as follows:

```
[Attributes]
AttributeName=
AttributeName=
```

.
. .
.

For example:

```
[Attributes]
User-Name=
NAS-IP-Port=
Service-Type=
Framed-Protocol=
Framed-IP-Address=
Framed-IP-Netmask=
```

The [Attributes] section lists one *AttributeName* on each line. You must ensure that an equal sign (=) immediately follows each *AttributeName*, with no spaces in between. Improperly formatted entries are considered invalid and are ignored.

Each *AttributeName* in the [Attributes] section must be defined in a dictionary file (.dct) installed on the Steel-Belted Radius server. This dictionary can be standard RADIUS or vendor-specific.

Note: The first five attributes in each authentication log file entry (Date, Time, RAS-Client, Full-Name, and ACC/REJ) are always enabled, and cannot be re-ordered or deleted. Therefore, these attributes do not appear in the authlog.ini file [Attributes] section.

authlog.ini [Configuration] Section

authlog.ini [Configuration]	
field	Meaning
LogDir	If this setting is present, it overrides the default system location (the private directory). You might need to add this section and field if it does not exist in your authlog.ini file. <i>NOTE: With directed realms, you can still maintain multiple authentication log locations.</i>

authlog.ini [Settings] Section

Steel-Belted Radius writes all authentication request data to the current authentication request log file (.authlog) until that log file is closed. When Steel-Belted Radius closes an authentication request log file, it immediately opens a new one and begins writing authentication request data to it.

You can configure how often this “rollover” of the authentication request log file occurs.

The naming conventions of the authentication request log files support the fact that Steel-Belted Radius can create more than one file per day. The formats are as follows. In the examples below, *y*=year digit, *m*=month digit, *d*=day digit, and *h*=hour digit. The extra sequence number *_nnnnn* starts at *_00000* each day.

File Generation Method	File Naming Convention
Default (24 hours)	<i>yyyymmdd.authlog</i>
Non-24-hour rollover	<i>yyyymmdd_hhmm.authlog</i>
Rollover due to size	<i>yyyymmdd_nnnnn.authlog</i>
Rollover due to size or startup when non-24-hour time in effect	<i>yyyymmdd_hhmm_nnnnn.authlog</i>

The following fields have been provided for use in the [Settings] section of *authlog.ini*. These fields control which entries are written to the authentication request log file, and ensure the compatibility of these entries with a variety of database systems. The following “rollover” fields can be present in the [Settings] section.

authlog.ini	
[Settings] field	Meaning
Enable	If set to 0 (the default), the authentication request log is disabled. If set to 1, the authentication request log is enabled. Authentication servers should have Enable set to 1; for efficiency, non-authentication servers should have Enable set to 0.
BufferSize	The size of the buffer used in the authentication request logging process, in bytes. The default is 32768.
LineSize	The maximum size of a single authentication request log line. The default is 4096. The allowable range is 1024 to 32768.
MaxSize	The maximum size of an authentication request log file, in bytes. Once the authentication request log file reaches this limit, the log file is closed and a new file started. A value of 0 (the default) means unlimited size.
QuoteBinary	If set to 1 (the default), binary values written to the authentication request log file are enclosed in quotes. If set to 0, quotes are not used. You should set this value according to the format expected by the application that eventually receives the authentication request log entries.

authlog.ini [Settings] field	Meaning
QuoteInteger	If set to 1 (the default), integer values written to the authentication request log file are enclosed in quotes. If 0, quotes are not used. You should set this value according to the format expected by the application that eventually receives the authentication request log entries.
QuoteIPAddress	If set to 1 (the default), IP addresses written to the authentication request log file are enclosed in quotes; if 0, quotes are not used. You should set this value according to the format expected by the application that eventually receives the authentication request log entries.
QuoteText	If set to 1 (the default), text strings written to the authentication request log file are enclosed in quotes; if 0, quotes are not used. You should set this value according to the format expected by the application that eventually receives the authentication request log entries.
QuoteTime	If set to 1 (the default), time and date values written to the authentication request log file are enclosed in quotes; if 0, quotes are not used. You should set this value according to the format expected by the application that eventually receives the authentication request log entries.
Rollover	How often the current authentication request log file is closed and a new file opened (a rollover), up to one rollover per minute. Non-zero values indicate the number of minutes until the next rollover. A value of 0 (the default) causes a rollover once every 24 hours, at midnight local time.
RolloverOnStartup	If set to 1, each time Steel-Belted Radius is started up, it closes the current authentication request log file and open a new one. A sequence number <i>_nnnnn</i> is appended to the log file name, just as when MaxSize is reached. If 0 (the default), each time Steel-Belted Radius is started up, it appends entries to the previously open authentication request log file.
Titles	If set to 1 (the default), each time a new authentication request log file is created, the title line (containing column headings) is written to the file. If 0, the line is not written.
UTC	If set to 1, time and date values are provided according to universal time coordinates (UTC, formerly known as Greenwich mean time or GMT). If 0 (the default), time and date values reflect local time.

blacklist.ini File

The blacklist.ini configuration file enables and configures blacklist settings. Only one profile can be created for the purposes of blacklisting, and all login attempts that match that profile are blocked. An authentication request matches the blacklist profile if the attributes in the request match the Check-List attributes of the profile. The profile can contain multiple attributes, and if any of the attributes match those of the profile, the attempt is rejected.

The blacklist.ini file contains one configuration section called [Settings], and has the following settings:

```
[Settings]
Enable = <0/1>
Profile = profile
IncludeProxy = <0/1>
```

where the fields have the following meanings:

blacklist.ini	
[Settings] Field	Meaning
Enable	If set to 1 (the default), blacklisting is enabled. If set to 0, the feature is disabled.
Profile	The name of the blacklisting profile in the Steel-Belted Radius database.
IncludeProxy	If set to 1 (the default), blacklisting is configured to include proxy requests - meaning, it is applied to all authentication requests. If set to 0, blacklisting is configured only to local authentication requests.

For example:

```
[Settings]
Enable = 1
Profile = BlockedNumbers
IncludeProxy = 0
```

You must now create a profile called `BlockedNumbers` (with the Administrator, for example) that contains Check-List attributes to match with. In this example, we might provide a list of `Calling-Station-Id` phone numbers to blacklist rogue users.

```
Calling-Station-Id = 617-999-9119
Calling-Station-Id = 800-515-7889
```

bounce.ini File (Windows only)

The bounce.ini configuration file enables and configures the Steel-Belted Radius auto-restart feature. This feature causes the Steel-Belted Radius server software to restart itself automatically whenever it experiences a shutdown.

If you enable the auto-restart feature, it results in the loading of two copies of the server executable, radius.exe. After the parent executable is run, the parent executable runs the child executable. The parent periodically sends a message to the child to see if it is still operating. If the child does not respond to the message within 60 seconds (a configurable time period), the parent terminates the child, waits for a configurable number of seconds to allow radius.exe to fully shut down, and then starts a new copy of the child.

Note: When auto-restart is enabled and the server is running normally, you should see two instances of radius.exe in any tool (such as the Task Manager) that you use to monitor processes on the Windows host computer.

While auto-restart is enabled, all server startup and shutdown activity is logged to a file called bounce.log in the server directory. Other types of server activity continue to be logged the all-purpose activity log file (yyyymmdd.LOG), also in the server directory.

The bounce.ini file contains one configuration section called [Settings]. The syntax for this section is as follows:

```
[Settings]
Enable = n
PingInterval = n
MaxPong = n
MaxStartup = n
MaxShutdown = n
```

where the fields have meanings as follows:

bounce.ini	
[Settings] Field	Meaning
Enable	If set to 1 (the default), the auto-restart feature is enabled. If set to 0, the feature is disabled.
MaxPong	The number of seconds the parent waits for a response message from the child, before it decides the child is no longer operating and attempts to restart it. The default is 17 seconds.

bounce.ini**[Settings] Field Meaning**

MaxShutdown	The number of seconds the parent allows for normal shutdown of the child. If the child does not terminate within that time, the parent terminates the child. The default is 20 seconds.
MaxStartup	The number of seconds the parent allows for starting up the child. If the child does not send a message within that time, the parent decides the startup was not successful and exits. The default is 60 seconds.
PingInterval	The number of seconds between each message sent by the parent to the child to check whether it is running. The default is 5 seconds.

Note: We strongly recommend that the default values remain unchanged in the bounce.ini file. If you decide to change any values, keep in mind that the MaxPong value should be greater than or equal to the PingInterval.

classmap.ini File

The classmap.ini initialization file specifies what Steel-Belted Radius does with original attributes encoded into the Class attribute included in accounting requests it receives.

classmap.ini [AttributeName] Section

The [AttributeName] section of classmap.ini specifies whether RADIUS information encapsulated in a Class attribute for an accounting request should be added to the accounting request or replace a current value in an accounting request. If one attribute is replaced by another, the original attribute can be added to the request with a different identifier.

```
[AttributeName]  
<add | replace> = Attribute [,Attribute]
```

Field	Meaning
<i>AttributeName</i>	Name of the attribute encoded into the Class attribute by the authenticating server.
<add replace>	Specifies whether the attribute value should be added to the accounting request (leaving all other values intact) or whether one value should replace another in the accounting request.
<i>Attribute</i>	Name of the attribute that should be added to the accounting request, which contains the original value of the attribute identified by <i>AttributeName</i> .
[, <i>Attribute</i>]	(Optional) Name of the attribute in the accounting request that should contain the value of the attribute displaced when <i>AttributeName</i> 's value replaced the existing <i>Attribute</i> value.

In the following example, the encapsulated User-name attribute would replace the existing User-Name in the accounting request.

```
[User-name]
Replace = User-Name
```

In the following example, the encapsulated User-Name attribute would be placed in the accounting request as User-Name and the original value for User-Name would be added to the request as Funk-Full-User-Name.

```
[User-name]
Replace = User-Name, Funk-Full-User-Name
```

In the following example, the encapsulated User-Name attribute would be added to the accounting request as a new attribute (the original User-Name attribute would remain unchanged).

```
[User-Name]
Add = Funk-Full-User-Name
```

dhcp.ini File

The dhcp.ini configuration file configures DHCP address pools so that IP addresses can be assigned from a backend DHCP server, rather than from a standard Steel-Belted Radius IP address pool.

dhcp.ini [Settings] Section

The following fields have been provided for use in the [Settings] section of the dhcp.ini file to control DHCP address allocation:

dhcp.ini [Settings] Field	Meaning
Enable	Set to 1 to enable DHCP address allocation, or set to 0 to disable it.
Attempts	Set to the number of times a DHCP DISCOVER or REQUEST message is sent if no response is received. The default value is 3.
AttemptTimeout	Set to the waiting period, in seconds, for a response to a DISCOVER or REQUEST message, before resending the message. The default value is 2 seconds.
OverallTimeout	Set to the number of seconds for acquiring an IP address before a failure. This timeout applies only to the DISCOVER/REQUEST sequence used to initially acquire an address, not to address renewal or release. <i>NOTE: while the timeout for the individual DISCOVER and REQUEST transactions is specified by Attempts and AttemptTimeout, Overall Timeout specifies the timeout for the entire sequence.</i> The default value is 10 seconds.
htype	The client hardware type (0 - 255). Normally, this field should be omitted, as the value is generated automatically.
Hlent	The length of the client hardware address (1 - 16.) Normally, this field should be omitted, as the value is generated automatically.
Chaddr-prefix	A string that specifies the initial bytes of the client hardware address (chaddr). This string can include escape codes, including \nnn for decimal values and \xnn for hex values. Normally, this field should be omitted, as the value is generated automatically.
ServerPort	Set to the UDP port number that the DHCP server(s) listen on. This field should be specified only for non-standard DHCP configurations. The default value is 67, which is the common DHCP server port.

dhcp.ini	
[Settings] Field	Meaning
LocalPort	Set to the UDP port number that Steel-Belted Radius, acting as a relay agent, uses during DHCP communication. This field should be specified for only non-standard DHCP configurations. The default value is 67, which is the standard DHCP server port.
Pad	Set to the minimum number of bytes for a DHCP request message. Messages smaller than this number are padded with 0s. Certain DHCP servers discard messages smaller than a certain value. This option allows interoperability with such servers. Default value is 0 (no padding.)

The following is a sample dhcp.ini file:

```
[Settings]
Enable = 1
Attempts = 3
AttemptTimeout = 2
OverallTimeout = 10
```

dhcp.ini [Pools] Section

The [Pools] section lists all DHCP pool names in the following format:

```
[Pools]
pool 1
pool 2
.
.
.
```

For example:

```
[Pools]
DHCP_SERVER1
DHCP_SERVER_SALES
```

pool.dhc Files

For each pool listed in the [Pools] section of the dhcp.ini file, there must be a corresponding *pool.dhc* file that configures that pool.

pool.dhc [Settings] Section

The following fields have been provided for use in the [Settings] section of the *pool.dhc* file:

<pool>.dhc	
[Settings] Field	Meaning
LeaseTime	Set to the lease time, in seconds, to request from the DHCP server. The default value is 1 day.
MinLeaseTime	Set to the minimum lease time, in seconds. Offers from DHCP servers with lease time less than this minimum are ignored. Default value is the value set for LeaseTime.
TargetAddress	Set to the address to which DISCOVER messages are sent. Default value is 255.255.255.255, the local broadcast address. This entry should normally be omitted, to allow DHCP DISCOVER messages to be broadcast.

pool.dhc [Request] Section

The [Request] section allows options in the DHCP DISCOVER and REQUEST messages to be constructed from attributes in the RADIUS Access-Request and from pre-configured literal values in the following way:

```
[Request]
  DHCP option = RADIUS attribute or literal value
  DHCP option = RADIUS attribute or literal value
  .
  .
  .
```

The *DHCP option* contains of the following fields (brackets ([])) indicate optional text). Fields are not separated by spaces.

```
[vendor-specific] option [offset] format
```

<DHCP option>	
Field	Meaning
vendor-specific	Set to <i>v</i> if this is a vendor-specific option, or omit otherwise.
option	Set to the DHCP option in the format, <i>nnn</i> .

<DHCP option>	
Field	Meaning
offset	Set to either a period ('.') followed by the number of bytes into the option where the value is located, or a plus-sign ('+') to indicate a list of values in the DHCP option -- each to be mapped to an instance of the RADIUS attribute.
format	Set to the format of the DHCP option, which can be one of the following: <ul style="list-style-type: none"> • n32 - a 32-bit integer • n16 - a 16-bit integer • n8 - an 8-bit integer • s or string - a string • i or ip - an IP address

The following are examples of *DCHP option* fields:

- 1ip (The "Subnet Mask" option as an IP address)
- 3+ip (The "Router" option as a list of IP address, each to be mapped to an instance of the RADIUS attribute)
- 6.4ip (The "DNS Server" option as a second IP address in list (each IP address is 4 bytes))
- 12s (The "Host Name" as a string)

The RADIUS attribute can be set to the name of any attribute defined in any dictionary. A literal value can be specified instead of a RADIUS attribute. This value must be text enclosed in double-quotes ("").

The string is interpreted based on the format of the DHCP option, according to the following:

- IP addresses must be specified in dotted notation - for example, "127.0.0.1"
- Integers are expressed in decimal format - for example, "100"
- Strings are expressed as any text sequence

The text can include escape sequences, where the backslash character ('\') is the escape character. Escape sequences are interpreted as follows:

Escape Code	Meaning
\a	7
\b	8

Escape Code	Meaning
\f	12
\n	10
\r	13
\t	9
\y	11
\nnn	A decimal value between 0 and 255.
\xnn	A hexadecimal value between 00 and FF
\\	A literal backslash '\'
\"	A double-quote
\char	A single character, interpreted literally

Note: You must use an escape character to include a literal backslash ('\') or double-quote ("") in the string.

An escape sequence can be used to set an option to an arbitrary binary value. This is useful, for example, when setting the Vendor Class Identifier option (60).

The following example sets the DHCP `Host Name` option to the RADIUS `Calling-Station-Id`, and sets the DHCP `Vendor Class Identifier` option to a binary string:

```
[Request]
12s = Calling-Station-Id
60s = "\x01\x02\x03\x04\x05"
```

***pool.dhc* [Reply] Section**

The [Reply] section allows RADIUS Access-Accept attributes to be constructed from options the DHCP server returns in an ACK message, in the following way:

```
[Reply]
RADIUS attribute = DHCP option
RADIUS attribute = DHCP option
.
.
.
```

The RADIUS attribute and the *DHCP option* are specified just as for the [Request] section.

Note: In contrast to the [Request] section, the left and right sides of the equal sign are reversed to account for the direction in which the data is being set.

The following example returns the RADIUS Framed-IP-Netmask attribute from the DHCP Subnet Mask option and sets the RADIUS Framed-MTU attribute from the DHCP Interface MTU option:

```
[Reply]
Framed-IP-Netmask = 1ip
Framed-MTU = 26n16
```

Reconfiguring Pools

DHCP pool information is loaded at startup from the `dhcp.ini` file and all associated `pool.dhc` files. DHCP pools can be added, deleted, and modified dynamically by doing the following:

- 1 Modify the `dhcp.ini` file and the `pool.dhc` files as required.
- 2 Depending on your platform:
 - Under **UNIX**: Issue the HUP signal to the Steel-Belted Radius process.
 - Under **Windows**: Run RADHUP.EXE from the command shell.

The modified files are re-read and the pool configuration reset appropriately.

events.ini File

The `events.ini` configuration file controls dilutions and thresholds for Steel-Belted Radius events which are used to communicate failures, warnings, and other information. Events are handled by the Windows NT Events mechanism.

See “Windows NT Events” on page 168 for a list of valid event values. Look for the number in the left-most column in the tables listing Informational, Warning, and Error events. Note that only some of these events support thresholds or are dilutable.

events.ini [EventDilutions] Section

The `[EventDilutions]` section of `events.ini` controls how often a Steel-Belted Radius event is actually generated, in relation to how often the event is detected by the system. This feature allows certain events (those being issued too frequently to make it practical to report each one of them) to be logged on a more infrequent basis.

The correct syntax is as follows:

```
[EventDilutions]
EventName=DilutionCount
```

where *EventName* identifies a Steel-Belted Radius event and *DilutionCount* specifies how many times this event must occur before it is recorded in the Windows NT event log or to the SNMP manager program.

The following example configures warning event number 5008:

```
[EventDilutions]
SQLConnectFailure=5
```

This example assumes Steel-Belted Radius is configured to authenticate against a SQL database; and after configuring the `radsq1.aut` file and restarting the server, some SQL error condition exists, preventing Steel-Belted Radius from successfully connecting to the database. Steel-Belted Radius continues to attempt the connection and reports these attempts in the `date.log` file. However, Steel-Belted Radius does not trigger warning event 5008 until the fifth connection attempt fails.

events.ini [Suppress] Section

The [Suppress] section of `events.ini` allows you to suppress Steel-Belted Radius events. For example:

```
[Suppress]
5010
5020
```

events.ini [Thresholds] Section

The [Thresholds] section of `events.ini` allows you to specify thresholds that trigger specific events. These settings can involve more than one event type.

This section allows for fine tuning of Steel-Belted Radius event generation in regards to crucial items such as system memory, thread count, and file system space, and can differ for each computer depending on resources, configuration, and other applications.

This table lists the event names for both Windows and UNIX, in that order.

The following fields can be present:

events.ini [Thresholds] Field	Meaning
<code>ThreadAvailWarningIssue=x</code>	When the number of available threads reaches <i>x</i> , issue the warning event <code>RADMSG_THREADS_LOW</code> or <code>funkSbrTrapLowThreads</code> (5001).

events.ini [Thresholds] Field	Meaning
ThreadAvailWarningClear= <i>y</i>	When the number of available threads reaches <i>y</i> at some later point, issue the informational event RADMSG_THREADS_NORMAL or funkSbrTrapThreadsNormal (102).
FileSystemFreeKBWarningIssue= <i>x</i>	When the number of kilobytes of available system disk space reaches <i>x</i> , issue the warning event RADMSG_FILE_SYSTEM_LOW or funkSbrTrapLowFSSpace (5007).
FileSystemFreeKBWarningClear= <i>y</i>	When the number of kilobytes of available system disk space reaches <i>y</i> at some later point, issue the informational event RADMSG_FILE_SYSTEM_NORMAL or funkSbrTrapFSNormal (103).
ReserveMemoryKB= <i>x</i>	Reserve this amount of memory (in kilobytes) for cases of overload.
PoolPctAddressAvailWarningIssue= <i>x</i>	When the number of available IP addresses in any IP Address Pool drops below, <i>x</i> %, issue a funkSbrTrapIPAddrPoolLow warning.
PoolPctAddressAvailWarningClear= <i>y</i>	If the number of available IP addresses in any IP Address Pool has fallen below <i>x</i> % but have returned to above <i>y</i> %, issue an informational message.

For example:

```
[Thresholds]
ThreadAvailWarningIssue=10
ThreadAvailWarningClear=20
```

Using this example, a warning event (5001) is issued when the number of available accounting or authentication threads fall below 10 percent, and an informational event (102) is issued when it rises above 20 percent.

filter.ini File

The filter.ini configuration file allows you to set up rules for filtering attributes into and out of RADIUS packets.

If you edit filter.ini, you can apply your configuration changes dynamically, without stopping the server. Depending on your operating system:

- Under **UNIX**: Simply issue the HUP signal to the Steel-Belted Radius process:

kill -HUP *ProcessID*

- Under **Windows**: Run the RADHUP.EXE program from the command shell. (RADHUP.EXE is located in the server directory that you specified at installation time, usually C:\RADIUS\Service.)

Note: Rarely, you must edit radius.ini while configuring a realm. If you do edit radius.ini, you must stop and restart the Steel-Belted Radius before your new configuration is fully loaded.

Filter Rules

Each filter in the filter.ini file consists of the filter name in square brackets (*[name]*) followed by the rules for that filter.

Each rule takes one of the following three forms:

```
keyword attribute value
keyword attribute
keyword
```

The complete set of valid syntax combinations is as follows:

filter.ini Rule Syntax	Meaning
ALLOW	This keyword by itself specifies that all attributes, regardless of value, are to be allowed in the packet.
ALLOW <i>attribute</i>	This rule specifies that this attribute is allowed in the packet, regardless of its value.
ALLOW <i>attribute value</i>	The rule lists a specific attribute/value pair to allow in the packet.
EXCLUDE	The keyword by itself specifies that all attributes, regardless of value, are to be excluded from the packet.
EXCLUDE <i>attribute</i>	The rule specifies that this attribute is excluded from the packet, regardless of its value.
EXCLUDE <i>attribute value</i>	The rule specifies an attribute/value pair to exclude from the packet.
ADD <i>attribute value</i>	The rule lists a specific attribute/value pair to add to the packet. The attribute is added after all other rules are processed.
REPLACE <i>attr1</i> WITH <i>attr2</i>	The rule specifies that any occurrence of <i>attr1</i> are replaced by <i>attr2</i> , which retains <i>attr1</i> 's value.
REPLACE <i>attr1</i> WITH <i>attr2 v2</i>	The rule specifies that any occurrence of <i>attr1</i> (regardless of value) is replaced by <i>attr2</i> whose value is set to <i>v2</i> .

filter.ini Rule Syntax	Meaning
REPLACE <i>attr1 v1</i> WITH <i>attr2</i>	The rule specifies that any occurrence of <i>attr1</i> whose value is <i>v1</i> is replaced by <i>attr2</i> (which keeps value <i>v1</i>).
REPLACE <i>attr1 v1</i> WITH <i>attr2 v2</i>	The rule specifies that any occurrence of <i>attr1</i> whose value is <i>v1</i> is replaced by <i>attr2</i> having a value <i>v2</i> .

An attribute is ADDED to a packet only if it is legal to do so. Some attributes can appear only once in a RADIUS packet; others can appear multiple times. Thus, if an attribute that is the subject of an ADD rule is already present in the packet (after processing ALLOW and EXCLUDE rules) and can only appear once, it is not added.

The Steel-Belted Radius dictionary file `radius.dct` provides string aliases for certain integer values defined in the RADIUS standard. You are free to use these strings in attribute filter rules.

Warning: You can set up filter rules in any way you want. While this provides you with tremendous flexibility, it also means that Steel-Belted Radius does not prevent you from creating an invalid RADIUS packet. Some attributes are not appropriate for certain types of requests. For example, adding a pooled Framed-IP-Address to an accounting request could cause a loss of available IP addresses.

Order of Filter Rules

The order of rules is important. General default rules that take no parameters, such as ALLOW (allow all attributes unless otherwise specified) or EXCLUDE (exclude all attributes unless otherwise specified) must appear as the first rule in the filter. Later rules supersede earlier rules; the last applicable rule “wins.” ADD rules are applied after the ALLOW and EXCLUDE rules.

More specific rules with more parameters (ADD *attribute value*) act as exceptions to less specific rules with fewer parameters (ALLOW *attribute*, EXCLUDE). For example, you might want to ALLOW a certain attribute and EXCLUDE one or more specific values for that attribute. Or you might EXCLUDE all attributes, ALLOW specific attributes, and ADD specific attribute/value pairs.

There are two basic approaches to designing a filter:

- Start the rule list with a default EXCLUDE rule (no parameters) and add ALLOW rules for any attributes or attribute/value pairs that you want to insert into the packet. ADD and REPLACE rules may also be used.

- Start the rule list with a default `ALLOW` rule (no parameters) and add `EXCLUDE` rules for any attributes or attribute/value pairs that you want to remove from the packet. `ADD` and `REPLACE` rules may also be used.

Values in Filter Rules

The value of an attribute is interpreted based on the type of the attribute.

Attribute Type	Meaning																						
hexadecimal	A hexadecimal value is specified just as a string. Special characters may be included using escape codes.																						
int1, int2, int4, integer	1-, 2- or 4-byte decimal number (integer is equivalent to int4). <i>NOTE: The Steel-Belted Radius dictionary file radius.dct provides string aliases for certain integer values defined in the RADIUS standard. You are free to use these strings in attribute filter rules.</i>																						
ipaddr, ipaddr-pool	An IP address in dotted notation; for example: <code>EXCLUDE NAS-IP-Address 127.0.0.1</code>																						
ipxaddr-pool	A sequence of hex digits; for example: <code>ALLOW Framed-IPX-Network 0042A36B</code>																						
string	String attribute (includes null terminator). A string is specified as text. The text may be enclosed in double-quotes (""). The text is interpreted as a regular expression. Backslash (\) is the escape character. Escape codes are interpreted as follows: <table border="0"> <thead> <tr> <th>Code</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td><code>\a</code></td> <td>7</td> </tr> <tr> <td><code>\b</code></td> <td>8</td> </tr> <tr> <td><code>\f</code></td> <td>12</td> </tr> <tr> <td><code>\n</code></td> <td>10</td> </tr> <tr> <td><code>\r</code></td> <td>13</td> </tr> <tr> <td><code>\t</code></td> <td>9</td> </tr> <tr> <td><code>\v</code></td> <td>11</td> </tr> <tr> <td><code>\nnn</code></td> <td><i>nnn</i> is a decimal value between 0 and 255</td> </tr> <tr> <td><code>\xnn</code></td> <td><i>nn</i> is a hexadecimal value between 00 and FF</td> </tr> <tr> <td><code>\c</code></td> <td><i>c</i> is a single character, interpreted literally</td> </tr> </tbody> </table> <p>Literal backslashes (\) within a string and double-quotes (") within quoted strings should be prefixed with an escape character. For example:</p> <pre>ADD Reply-Message Session limit is one hour ADD Reply-Message "Session limit is one hour" ADD Reply-Message "Your user name is \"George\""</pre>	Code	Meaning	<code>\a</code>	7	<code>\b</code>	8	<code>\f</code>	12	<code>\n</code>	10	<code>\r</code>	13	<code>\t</code>	9	<code>\v</code>	11	<code>\nnn</code>	<i>nnn</i> is a decimal value between 0 and 255	<code>\xnn</code>	<i>nn</i> is a hexadecimal value between 00 and FF	<code>\c</code>	<i>c</i> is a single character, interpreted literally
Code	Meaning																						
<code>\a</code>	7																						
<code>\b</code>	8																						
<code>\f</code>	12																						
<code>\n</code>	10																						
<code>\r</code>	13																						
<code>\t</code>	9																						
<code>\v</code>	11																						
<code>\nnn</code>	<i>nnn</i> is a decimal value between 0 and 255																						
<code>\xnn</code>	<i>nn</i> is a hexadecimal value between 00 and FF																						
<code>\c</code>	<i>c</i> is a single character, interpreted literally																						

Attribute Type	Meaning
time	<p>A time value is specified with a string indicating date and time: <code>yyyy/mm/dd hh:mm:ss</code></p> <p>The date portion is mandatory; the time portion may be specified to whatever degree of precision is required, or may be omitted entirely. For example: <code>2003/7/3 14:00:00</code> and <code>2003/7/3 14</code> both refer to July 3, 2003 at 2:00 p.m. For example: <code>ADD Ascend-PW-Expiration 2003/7/1</code></p>

Referencing Attribute Filters

Steel-Belted Radius attribute filtering offers total flexibility. You can use the same filter for all packets in all realms. You can apply filtering to some realms, and not others (to disable filtering for a realm, omit filtering parameters from the `.pro`, `.dir` or `ttlsauth.aut` file). Filtering is typically used only for packets that are routed “out” to realms (the `FilterOut` parameter).

To reference the filtering rules defined in the `filter.ini` file, you must use the `FilterOut` and `FilterIn` parameters in the `[Auth]` and `[Acct]` sections of a RADIUS realm configuration file.

The full syntax used is:

```
[Auth]
FilterIn=name1
FilterOut=name2

[Acct]
FilterIn=name3
FilterOut=name4
```

where `name1`, `name2`, and so forth provide the names of filters, sections in the `filter.ini` file called `[name1]`, `[name2]`, and so forth. The `name` values in this syntax are completely independent of each other. They may be all the same, all different, or some combination of same and different.

Important: *If a `[name]` section is not found in the `filter.ini` file, it is equivalent to assigning a filter that EXCLUDEs all attributes. In other words, assigning a filter name that cannot be found causes the final packet to be emptied of all attributes.*

Important: In accounting filters, you should not allocate IP addresses from Steel-Belted Radius IP address pools, as these addresses are never released.

Note: When using the `FilterIn` and `FilterOut` parameters in the `[Auth]` and `[Acct]` sections, be sure to use the filter name without the square brackets ("`name`", not "`[name]`").

lockout.ini File

The `lockout.ini` configuration file enables and configures lockout settings. The `lockout.ini` file contains one configuration section called `[Settings]`, and has settings similar to the following:

```
[Settings]
Enable = 0
Rejects = 3
Within = 120
Lockout = 600
```

where the fields have the following meanings:

lockout.ini	
[Settings] Field	Meaning
Enable	If set to 0 (the default), lockout is enabled. If set to 1, lockout is disabled.
Lockout	The lockout period in seconds.
Rejects	The number of rejected attempts prior to lockout.
Within	The period in seconds during which a specified number of rejects causes a lockout.

Clearing Locked-Out Accounts

To clear a locked-out account, use one of the following methods:

- Allow the lockout period to expire.
- Using the LDAP Configuration Interface, create and execute an LDIF file with the following commands:

```
dn: user=user_name,radiusstatus=lockout,o=radius
changetype: delete
```

where ***user_name*** is the name of the locked out user.

For information on using the LDAP configuration interface, see “LDAP Configuration Interface” on page 332.

radius.ini File

The radius.ini initialization file is the main configuration file that determines the operation of the Steel-Belted Radius server. It contains information that controls a variety of server functions, primarily authentication.

Warning: Use caution when editing radius.ini, so that values pertaining to one feature are not overwritten or lost while you configure another feature. You should make a backup copy of radius.ini before you make changes.

radius.ini [Addresses] Section

If you are using a server that has more than one network interface (a multi-homed server), you may need to guarantee which interfaces are bound and which are ignored by naming them explicitly. To do this, you must add an [Addresses] section to the radius.ini file.

You must choose whether you want all of your proxy traffic routed through one interface, or if you want to dedicate interfaces to particular realms.

See “proxy.ini [Interfaces] Section” on page 274 if you want to configure your system to route proxy realms through specific interfaces.

To route all of your proxy traffic through a single interface, set the value for ProxySource (in the [Configuration] section of radius.ini) to the appropriate IP address(es) (which must have been listed in the [Addresses] section).

For example:

```
[Addresses]
192.10.20.30
192.10.20.31

[Configuration]
ProxySource = 192.10.20.30
```

radius.ini [Certificate] Section

The [Certificate] section of radius.ini specifies information about the server's certificate and private key, which are required by the EAP-TLS, EAP-TTLS, and EAP-PEAP plug-ins.

The [Certificate] section consists of the following fields:

[Server_Settings] field	Meaning
Server_Certificate_Info_File	The full path of the file that contains information about the server's certificate. This is not the location of the PKCS#12 file that contains the certificate, but rather the file that contains information about it.

For example, the [Certificate] section might look like this for a Solaris host:

```
[Certificate]
Server_Certificate_Info_File = /usr/local/radius/certInfo.ini
```

Similarly, the [Certificate] section might look like this for a Windows host:

```
[Certificate]
Server_Certificate_Info_File = c:\radius\service\certInfo.ini
```

Server Certificate Info File

The server certificate information file is an ASCII file with a single section, [Certificate_Info]. This separate file is meant to allow the administrator to isolate it in a portion of the file system that would be accessible to the Steel-Belted Radius server process but not to general users or operators of the system.

Server Certificate Info file	
[Certificate_Info] field	Meaning
Certificate_And_Private_Key_File	Identifies the PKCS#12 file containing the server's certificate (chain) and private key. This should be specified as a complete file system path (though on Solaris, relative paths work as well) to remove any ambiguity regarding the file being used.
Password	Specifies the password required to retrieve the server's private key that was included in the PKCS#12 file.

Example

```
[Certificate_Info]
; Location of the PKCS#12 file containing the certificate
```

```

; and private key of the server and all certificates necessary to
; establish a chain to the Certificate Authority that issued
; the certificate.
Certificate_And_Private_Key_File = c:\radius\service\test_server.pfx

; Password with which the private key contained in the PKCS#12
; file mentioned above was encrypted.
Password = tryme

```

Note: Setting the ProxySource value overrides proxy routing on a realm-by-realm basis.

radius.ini [AuthRejectLog] Section

The authentication log (described in “Authentication Log File” on page 146), which supplements the radius log file (described in “Radius Log File” on page 144), records the reason for authentication rejections, which the radius log file does not record. You configure the [AuthRejectLog] section of radius.ini to specify what types of authentication rejection messages Steel-Belted Radius records in the authentication log. You can specify that you want the authentication log to record all reject messages, reject messages of one or more specific types, or the most relevant rejection messages.

Processing an authentication request may result in multiple instances of an authentication method being given a chance to authenticate the user. If this occurs and at least one authentication method succeeds in authenticating the user, no messages are recorded to the log file. If this occurs and all instances fail to authenticate the user, you can specify that only the most relevant message is recorded. For example, if one instance resulted in an authentication error of type `InvalidCredentials` and another results in an authentication error of type `SystemError`, only the `InvalidCredentials` message would be logged.

You can specify that more than one type of log message should be recorded by entering more than one filter type value for the Filter field.

radius.ini	
[AuthRejectLog] Field	Meaning
Enable	If set to 0 (the default), authentication reject messages are not recorded in the log file. If set to 1, authentication reject messages of the specified type(s) are recorded in the log file.

radius.ini**[AuthRejectLog] Field Meaning**

Filter	<p>Specifies the types of authentication reject messages to be recorded:</p> <ul style="list-style-type: none">• <code>All</code> – Record all authentication rejection messages.• <code>MostRelevant</code> – When multiple authentication methods are tried and all fail, record the most relevant error messages (that is, the messages with the greatest severity). If two messages have the same severity, both are listed. <p>The following values are listed in order of greatest to least relevance:</p> <ul style="list-style-type: none">• <code>PostProcessRejection</code> – User was authenticated successfully but postprocessing caused rejection.• <code>InvalidCredentialsOrUser</code> – User was not authenticated because user was not found or credentials were invalid.• <code>InvalidCredentials</code> – User was not authenticated because user was known but the password or certificate was not correct.• <code>UnsupportedCredentialType</code> – User was not authenticated because the credentials presented were of the wrong type.• <code>UserNotFound</code> – User was not authenticated because user could not be found in the authentication database.• <code>AccessError</code> – Authentication failed because a database or remote server was inaccessible.• <code>InvalidRequest</code> – User was not authenticated because the request appeared to be malformed.• <code>BlacklistedUser</code> – User was not authenticated because user is blacklisted.• <code>SystemError</code> – User was not authenticated because of a system error such as a resource allocation error.
--------	---

For example, the following example would cause all authentication log messages to be recorded.

```
[AuthRejectLog]
Enable = 1
Filter = All
```

The following example would cause all authentication log messages of type `SystemError` to be recorded.

```
[AuthRejectLog]
Enable = 1
```

```
Filter = SystemError
```

The following example would cause all authentication log messages of type `SystemError`, `BlacklistedUser`, or `UserNotFound` to be recorded.

```
[AuthRejectLog]
Enable = 1
Filter = SystemError, BlacklistedUser, UserNotFound
```

radius.ini [Configuration] Section

The [Configuration] section of `radius.ini` contains parameters that control the most basic behavior of the Steel-Belted Radius server. The following fields may be present:

radius.ini [Configuration] Field	
AcctAutoStopEnable	If set to 1, the Proxy AutoStop feature is enabled. If this field is not present, it is enabled by default.
AddSourceIPAddressAttrToRequest	If set to 0 (the default), Steel-Belted Radius does not add source address information to RADIUS requests. If set to 1, Steel-Belted Radius adds a <code>Funk-Source-IP-Address</code> attribute identifying the IP address from which the RADIUS request was received to the end of the collection of attributes in the packet. All processing that could be performed on an attribute included in the request packet, such as checklist processing, can be performed on this attribute. Note that, if you enable this attribute, the attribute is visible to the proxy module. If your environment proxies requests, you may want to configure Steel-Belted Radius to strip the attribute from the request before forwarding it to a downstream server.
Allow-Unmasked-Password	If set to <code>Yes</code> , it is possible through the Administrator program to view previously entered passwords (provided they are not strongly encrypted). If set to <code>No</code> (the default), it is possible to unmask passwords as you enter them, but not to view passwords that have already been entered.

radius.ini [Configuration] Field

Allow-Unmasked-Secret	<p>If set to <code>Yes</code>, it is possible through the Administrator program to view previously entered shared secrets.</p> <p>If set to <code>No</code> (the default), it is possible to unmask shared secrets as you enter them, but not to view shared secrets that have already been entered.</p>
Apply-Login-Limits	<p>If set to <code>Yes</code> (the default), then the maximum number of concurrent connections for each user is enforced, and connection attempts above the limit are rejected.</p> <p>If set to <code>No</code>, then connections above the limit are allowed, but an event is noted in the Radius log file.</p>
AttributeEdit	<p>If set to 1, this field enables the attribute editing feature for Proxy RADIUS realms.</p> <p>If set to 0, the feature is disabled. If the AttributeEdit field is not present, the feature is enabled by default.</p>
AuthenticateOnly	<p>If set to 1, no response attributes are included in the response packet to an AuthenticateOnly (Service-Type 8) request.</p> <p>If set to 0, the normal response attributes are included in the response.</p> <p>The default is 1.</p>
AutoPasswords	<p>If set to <code>Yes</code>, auto passwords for authentication against the native database are enabled. By default, this is disabled.</p>
CheckMessageAuthenticator	<p>If set to 1, the validation of received Message-Authenticator attributes is enabled.</p> <p>The default is 0.</p> <p>The validation of Message-Authenticator can occur either on receipt of an Access-Request from a NAS device or on receipt of an Access-Request, Access-Reject, or Access-Challenge from a proxy (extended proxy only).</p> <p><i>NOTE: validation does not occur for ordinary proxy.</i></p>

radius.ini [Configuration] Field

ClassAttributeStyle	<p>If set to 1, Steel-Belted Radius uses unencrypted Class attributes with multiple ASCII keys in Access-Reply packets.</p> <p>If set to 2 (the default), Steel-Belted Radius uses enhanced/encrypted Class attributes in Access-Reply packets. ClassAttributeStyle must be set to 2 if you are want to record accounting information for users authenticated with EAP-TTLS or EAP-PEAP.</p>
ExtendedProxy	<p>If set to 1, this field enables you to set up realms for Proxy RADIUS or directed authentication/accounting.</p> <p>If set to 0, Steel-Belted Radius can still proxy-forward to specific servers (identified using Proxy entries in the Administrator program), but Proxy RADIUS realms and directed realms are disabled.</p> <p>If the ExtendedProxy field is not present in the [Configuration] section, realms are disabled by default.</p> <p>See “Configuring a Proxy RADIUS Realm” on page 257.</p>
FramedIPAddressHint	<p>If set to <i>Yes</i>, the attribute Framed-IP-Address is treated as a hint. If this attribute appears in the Access-Request and the user’s return list is configured to allocate Framed-IP-Address from a pool, the IP address in the Access-Request is returned instead of a newly-allocated IP Address. See “Hints” on page 76.</p>
LogDir	<p>Sets the destination directory where Radius log files is stored.</p> <p>The default is the RADIUS database private directory.</p>
LogLevel	<p>Sets the rate at which Steel-Belted Radius writes entries to the radius log file (.LOG). The LogLevel may be the number 0, 1, or 2:</p> <ul style="list-style-type: none">• 0 is the production logging level• 1 is the informational logging level• 2 is the debug logging level <p>The LogLevel setting is re-read whenever the server receives a HUP signal.</p>

radius.ini [Configuration] Field

LogAccept	Specifies whether messages associated with Accepts that meet the current LogLevel should be recorded in the log file. It is set to 1 (on) by default. The LogAccept setting is re-read whenever the server receives a HUP signal.
LogReject	Specifies whether messages associated with Rejects that meet the current LogLevel should be recorded in the log file. It is set to 1 (on) by default. The LogReject setting is re-read whenever the server receives a HUP signal.
PhantomTimeout	The maximum number of seconds that a phantom accounting record remains active. As soon as the corresponding start packet is received, a phantom record is discarded. If a phantom record still exists at the end of its timeout period, it is discarded. See “Phantom Records” on page 80.
PrivateDir	Name of the location of the Steel-Belted Radius server directory; the server directory contains the database and dictionary files (if not specified, defaults to the same directory where the Steel-Belted Radius service/daemon resides).
ProxyFastFail	An amount of time, in seconds. The proxy fast-fail feature saves a Steel-Belted Radius server from continuing to forward packets to a Proxy RADIUS target that appears to be down temporarily. A reasonable value for ProxyFastFail might be 1800 (30 minutes). A value of 0 disables the feature. See “Proxy Fast-Fail” on page 69.
ProxySource	The IP address listed in the [Addresses] section corresponding to the interface through which all outgoing proxy traffic is routed.
ProxyStripRealm	This setting controls whether the proxy realm decoration is stripped before sending the request downstream. If set to 0, no realm name stripping is performed. The default is 1.

radius.ini [Configuration] Field

TraceLevel	The RADIUS packet tracing level between 0 and 2, where 0 indicates the default action of no packet tracing, 1 indicates that the parsed contents of packets is to be logged and 2 indicates that the raw contents of the packet is to be logged. Packet traces are written to the log file and can be a useful tool for troubleshooting interoperability problems.
TreatAddressPoolsAsDisjoint	If set to 1, then Steel-Belted Radius treats each IP address pool as though it operates off its own disjoint address space. Thus, this disables the normal checks to ensure that an IP address is allocated only to a single address pool. The default is 0, so that a single IP address can be allocated only to a single session and a single IP address pool. NOTE: Due to the requirements of resource management, Steel-Belted Radius uses the Class attribute to track IP addresses. This attribute contains the IP pool name and IP address.
UseNewAttributeMerge	If set to 1, the new profile and user attribute merging calculation is performed. If set to 0, the older calculation technique is used. See “Resolving profile and User Attributes” on page 59 for explanation of new attribute merge. Default is 1 (enabled).

radius.ini [CurrentSessions] Section

The [CurrentSessions] section of radius.ini controls the Current Sessions List. The following field may be present:

radius.ini[CurrentSessions] Field	Meaning
CaseSensitiveUsernameCompare	If set to 1 (the default), when the server searches its Current Sessions List for sessions that have the same username, it uses case-sensitive look-ups. If 0, it ignores case.

radius.ini [FailedAuthOriginStats] Section (Windows only)

The [FailedAuthOriginStats] section of radius.ini enables you to identify when a specific NAS is associated with certain Windows NT Performance Monitor

(perfmon) counters. This in turn helps you to identify a specific region of your network that may be having difficulties. The syntax is as follows:

```
[FailedAuthOriginStats]
RADIUSclient=IDnumber
RADIUSclient=IDnumber
.
.
.
```

where *RADIUSclient* is the name of a NAS or other client device as defined in the RAS Clients dialog, and *IDnumber* is a number in the range 1 to 16. These numbers map to the following Steel-Belted Radius perfmon counters:

```
Failed Auths - 1
Failed Auths - 2
.
.
.
Failed Auths - 16
```

For example, if you map a NAS named herman to the number 3:

```
[FailedAuthOriginStats]
herman=3
```

Then the perfmon counter `Failed Auths - 3` tells you the number of failed authentication requests that have originated from NAS `herman`.

See “Windows NT Performance Monitor” on page 163.

radius.ini [IPPoolSuffixes] Section

The [IPPoolSuffixes] section of radius.ini allows you to define suffixes that can be used to split the NAS-Specific IP Address Pools into smaller subcategories.

See “NAS-Specific IP Address Pools” on page 123.

The syntax is as follows:

```
[IPPoolSuffixes]
Suffix1
Suffix2
...
```

For example, to create three categories that append `-Bronze`, `-Silver`, and `-Gold` to IP Address Pool names, this section would be defined as follows:

```
[IPPoolSuffixes]
-Bronze
-Silver
```

radius.ini [LDAP] Section

The [LDAP] section of radius.ini sets the TCP port number that you want to use for communication between the LDAP server portion of Steel-Belted Radius and any LDAP clients.

See “LDAP Configuration Interface” on page 332.

The syntax is as follows:

```
[LDAP]
TCPport = value
```

where *value* is the port number that you want to use. The default port number is 667.

radius.ini [LDAPAddresses] Section

The [LDAPAddresses] section of radius.ini allows you to specify which interfaces Steel-Belted Radius listens on for LCI requests. If you want to provide these settings, you must add a section called [LDAPAddresses] to the radius.ini file. This section should contain a list of IP addresses, one per line.

For example:

```
[LDAPAddresses]
199.198.197.196
196.197.198.199
```

If the [LDAPAddresses] section is omitted or empty, Steel-Belted Radius listens for LCI requests on all bound IP interfaces.

Note: The LDAP Configuration Interface is an optional add-on for the Enterprise edition of Steel-Belted Radius. You must license the LDAP Configuration Interface before you can configure or use it.

radius.ini [NTDomain] Section (Windows only)

The [NTDomain] section of radius.ini configures the server’s response to an expired Domain password. You can choose separate responses for Domain User and Domain Group authentication methods. Steel-Belted Radius takes the actions that you define in the [NTDomain] section when it receives either of the following status codes after passing a username/password pair to an NT Domain for authentication:

- Expired password

- User must change password at next logon

The following fields may be present in a [NTDomain] section:

radius.ini [NTDomain] Field	Meaning
AllowExpiredPasswordsForUsers	<p>A value of <i>yes</i> means that when the incoming username/password pair can be validated but the password has expired (under Domain User authentication), the server responds with an Access-Accept. A value of <i>no</i> means the server responds with an Access-Reject.</p> <p>If you set this field to <i>yes</i>, and do not provide a ProfileForExpiredUsers value, the Access-Accept response contains the Return-List from the Domain User entry that matches the incoming username. This option is recommended for Domain Users.</p> <p>If you set this field to <i>yes</i>, and provide a ProfileForExpiredUsers, the Return-List from that profile are used.</p>
AllowExpiredPasswordsForGroups	<p>A value of <i>yes</i> means that when the incoming username/password pair can be validated but the password has expired (under Domain Group authentication), the server responds with an Access-Accept. A value of <i>no</i> means the server responds with an Access-Reject.</p> <p>If you set this field to <i>yes</i>, and provide a ProfileForExpiredGroups, the Return-List from that profile are used. This option is strongly recommended if you allow expired passwords for Domain Groups.</p> <p>If you set this field to <i>yes</i> and do not provide a ProfileForExpiredGroups value, the Access-Accept response contain the Return-List from the first Domain Group entry (alphabetically) in the server database.</p>
PrequalifyCheckList	<p>A value of <i>yes</i> means that Steel-Belted Radius performs checklist processing on each domain object in the database before trying to authenticate a user request. If checklist processing fails, the object is skipped.</p> <p>A value of <i>no</i> (default) means that prequalification checklist processing is not performed.</p> <p>For more information on prequalification checklist processing, see "Prequalification Checklists" on page 92.</p>

radius.ini [NTDomain] Field	Meaning
ProfileForExpiredUsers	Names a profile entry in the Steel-Belted Radius database. This entry provides the Return-List for responses for all users who are accepted by the server under Domain User "expired password" conditions.
ProfileForExpiredUsersInGroups	Names a profile entry in the Steel-Belted Radius database. This entry provides the Return-List for responses for all users who are accepted by the server under Domain Group "expired password" conditions. This option is strongly recommended for Domain Groups.

radius.ini [Ports] Section

The [Ports] section of radius.ini provides an alternative method for setting the UDP ports used by Steel-Belted Radius. The following fields may be present:

radius.ini [Ports] Field	Meaning
UDPAuthPort	The UDP port(s) used for authentication (one line per port assignment).
UDPAcctPort	The UDP port(s) used for accounting (one line per port assignment).

For example:

```
[Ports]
UDPAuthPort = 1645
UDPAuthPort = 1812
UDPAcctPort = 1646
UDPAcctPort = 1813
```

Warning: Be careful when configuring these settings and consider the impact on the /services file.

See “services File” on page 246.

Listening on Multiple UDP Ports

The following explains how the server determines which ports to listen on for incoming authentication requests (the logic for accounting requests works in the same manner):

- 1 If one or more UDPAuthPort settings are present in the [Ports] section of the radius.ini, the collection of unique port numbers specified in this section

represents the sum total of the ports on which the server listens for authentication requests.

A limit is imposed on the maximum number of ports that can be specified: the number of ports to listen on multiplied by the number of interfaces on the local host cannot exceed an operating-system-specific number. The total number of ports available on **Windows** computers is 64 and on **UNIX** computers is 4096, but some of these ports are already being used by other services when Steel-Belted Radius begins. If this limit is exceeded, the RADIUS authentication subcomponent fails to initialize.

- 2 If no UDPAuthPort settings are present in the [Ports] section, the server attempts to read the port number associated with the `radius` service specified in `/etc/services`. If present, the server listens on this port number.

Only one port can be specified in the services file. If multiple ports are required, they must be specified in the radius.ini file.

- 3 If no UDPAuthPort settings are present in the [Ports] section and no `radius` service is listed in the `/etc/services` file, the server listens for authentication requests on the default UDP ports 1645 and 1812.

The UDPActPort setting performs the same purposes for purposes of accounting, and the above information is relevant in the same way. The name of the service to look for in `/etc/services` is `radacct` and the fallback UDP ports are 1646 and 1813.

Note: Any failure to bind to one of the selected UDP ports causes the affected subcomponent (authentication or accounting) to fail to initialize.

radius.ini [SecurID] Section

The [SecurID] section of `radius.ini` contains items specific to SecurID authentication for ISDN users. It provides information that allows Steel-Belted Radius to briefly cache the user's credentials following a successful SecurID authentication. This technique is necessary to permit a second ISDN B-channel to be authenticated during the user's session. Steel-Belted Radius uses the cached token to authenticate the second channel.

Note: If this feature is not enabled, users who want to authenticate against a ACE/Server database via an ISDN connection that "bonds" both B-channels will fail to authenticate due to a SecurID security violation. ISDN users running only one B-channel are not affected.

The following fields may be present in a [SecurID] section:

radius.ini	
[SecurID] Field	Meaning
CachePasscodes	A value of <code>yes</code> enables the feature; no disables it.
SecondsToCachePasscodes	The number of seconds to retain the cached SecurID credentials (PIN and token code).

radius.ini [SecurID] Section - Enhanced Token Caching

This section pertains only to Ericsson equipment that supports enhanced token caching.

Enhanced token caching allows the administrator to specify that particular users are authenticated with both an ordinary password and a SecurID passcode. For such users, the ordinary PAP or CHAP password is checked first. If this first authentication is successful, the user's SecurID passcode is authenticated. Only if both authentications succeed is the user allowed access.

Note: The enhanced token caching feature does not interact in any way with the ordinary token caching feature described in the above section. Enhanced token caching is required to support the newer firmware releases on Ericsson devices.

Enhanced Token Caching Configuration

To enable enhanced token caching, a file with extension `.aut` must be included in the server directory - typically named `sidalt.aut`. It has the following format. (A slash inside angle brackets, such as `<0/1>`, indicates that one of the two values listed may be entered.)

```
[Bootstrap]
LibraryName = sidalt.dll
Enable = <0/1>
InitializationString = string

[Settings]
TokenAttr = string
CacheTimeoutAttr = string
MessageID = <0/1>
ChallengeTokenInPassword = <0/1>
```

The following table describes each field in the `sidalt.aut` file:

Sidalt.aut Field	Meaning
Enable	Set to 0 to disable, or 1 to enable.
InitializationString	Identifies the enhanced token caching software component for logging purposes. A typical value might be <code>SecurID Alt</code> . This field is required.
TokenAttr	The name of the Access-Request attribute containing the passcode or other information to be passed to the ACE/Server. This attribute must match the corresponding dictionary (<code>.dct</code> file) entry; that is, <code>Acc-Ace-Token</code> . This field is required.
CacheTimeoutAttr	The name of the Access-Response attribute containing the number of seconds a passcode remains in the cache from the time it was first validated by the ACE/Server. This attribute must match the corresponding dictionary (<code>.dct</code> file) entry; that is, <code>Acc-Ace-Token-Ttl</code> . This field is required.
MessageID	Controls the format of Reply-Message attribute response packets, which are used to prompt the user for information during a challenge, and to inform the user of results. Set to 0 to include only a text message. Set to 1 to include a 1-byte message ID followed by a text message. Message IDs are enumerated in the <code>messageids.h</code> file. If the <code>MessageID</code> entry is not present, its default value is 0.
ChallengeTokenInPassword	Allow test clients to interpolate with the enhanced token caching plug-in. If set to 1, passcode or other information entered by the user as a response to a challenge appears in the User-attribute, rather than in the attribute specified by <code>TokenAttr</code> entry. For internal use only.

Enhanced Token Caching Administration

To authenticate a user through the enhanced token caching component, the following must be true:

- 1 The attribute specified by the `TokenAttr` entry must be present in the Access-Request;
- 2 The Return-List specified for that user, either directly or through a profile, must include the attribute specified by the `CacheTimeoutAttr` entry.

If either attribute is not supplied, the user is assumed not to require enhanced token caching authentication, and is accepted.

Note: See your Ericsson product documentation for information about NAS and PPP client operation under the enhanced token caching authentication method.

radius.ini [Self] Section

The [Self] section of radius.ini lists all the realm names that indicate this Steel-Belted Radius server. The syntax is as follows:

```
[Self]
RealmName
RealmName
.
.
.
```

You can use the [Self] section to map a realm name to the Steel-Belted Radius server. This way, if you acquire a batch of new user accounts, users don't have to change anything about the way they enter usernames. They can enter the name *User<Delimiter>RealmName* or *RealmName<Delimiter>User* as usual.

When a username comes into Steel-Belted Radius, if the [Self] section lists *RealmName*, Steel-Belted Radius understands that it is the target, and handles the request locally instead of directing the request elsewhere.

See “User-Names with Multiple Suffix Delimiters” on page 62, “User-Names with Multiple Prefix Delimiters” on page 63 and “Proxy RADIUS” on page 65.

radius.ini [StaticAcctProxy] Section

The [StaticAcctProxy] section of radius.ini controls the delivery of Accounting messages to additional RADIUS Accounting-enabled devices on the network, even when the initial RADIUS transaction is not a Proxy-RADIUS transaction. The syntax is as follows:

```
[StaticAcctProxy]
Target = Name_ofSBR_dB_Proxy_entry
```

where *Name_ofSBR_dB_Proxy_entry* identifies the additional RADIUS Accounting-enabled device by name.

radius.ini [Strip] Section

The [Strip] section specifies if, and how, User-Name stripping is to occur. That is, these sections configure Steel-Belted Radius to manipulate the username by

stripping the incoming `User-Name` attribute value of realm names and other “decorations.”

The `[Strip]` section (and accompanying `[StripPrefix]` and `[StripSuffix]` sections) look like the following:

```
[Strip]
Authentication=Yes
Accounting=No
StripPrefixCharacters=@#%
StripSuffixCharacters="! "

[StripPrefix]
PrefixStringToStrip1
PrefixStringToStrip2
.
.
.
[StripSuffix]
SuffixStringToStrip1
SuffixStringToStrip2
.
.
.
```

The meaning of the `[Strip]` fields are as follows:

radius.ini	
[Strip] fields	Meaning
Authentication	If set to YES, then the <code>[StripPrefix]</code> and <code>[StripSuffix]</code> rules are used to strip the username before an authentication request is processed. The default is NO.
Accounting	If set to YES, then the <code>[StripPrefix]</code> and <code>[StripSuffix]</code> rules are used to strip the username before an accounting request is processed. The default is NO.
Proxy	Reserved for future use.
StripPrefixCharacters	A list of ASCII characters to strip from the prefix. If a space character appears in the list, the entire list must be surrounded by quotation marks.
StripSuffixCharacters	A list of ASCII characters to strip from the suffix. If a space character appears in the list, the entire list must be surrounded by quotation marks.

Strip Characters (Example)

Take the following [Strip] section as an example:

```
[Strip]
Authentication=yes
Accounting=yes
Proxy=no
StripPrefixCharacters = @#%
StripSuffixCharacters = " !"
```

If the incoming username is "@@@testuser !", then the string is processed and converted to "testuser".

Strip Prefix and Suffix

The [StripPrefix] section should give a list of any prefixes that should be removed from the beginning of usernames, including the delimiter.

The [StripSuffix] section should give a list of any prefixes that should be removed from the beginning of usernames, including the delimiter.

For example:

```
[Strip]
Authentication=yes
Accounting=yes
Proxy=no

[StripPrefix]
(isp.com\
(att.net])

[StripSuffix]
(@myrealm.com)
(@yahoo.com)
```

In this example, Steel-Belted Radius would strip the prefixes `isp.com\` or `att.net]` from usernames in authentication and accounting requests. Similarly, Steel-Belted Radius would strip the suffixes `@myrealm.com` or `@yahoo.com` in from usernames authentication or accounting requests.

Note: The [Strip] name processing does not apply to the LDAP plug-in.

radius.ini [ValidateAuth] and [ValidateAcct] Sections

The [Validate] sections in radius.ini allow username validation to occur. These sections enable Steel-Belted Radius to examine the User-Name attribute in the

incoming packet to determine whether it employs a valid character set, and to act accordingly. The following [Validate] sections are available:

```
[ValidateAuth]
User-Name = RegularExpression
```

```
[ValidateAcct]
User-Name = RegularExpression
```

The following fields may be present:

radius.ini																			
[Validate] Field	Meaning																		
[ValidateAuth]	This section applies only to authentication servers.																		
[ValidateAcct]	This section applies only to accounting servers.																		
User-Name	Names the regular expression against which the User-Name attribute is validated. If the User-Name entry is absent from the section or the regular expression is blank, no validation occurs.																		
RegularExpression	The regular expression lists each valid character or range of characters. A dash ('-') indicates a range of alphanumeric characters. For example, A-Z indicates every uppercase alphabetic character. A backslash ('\') followed by a non-alphanumeric character indicates that character literally, for example \? indicates the question mark. '\' is also used as an escape character, as follows: <table><tbody><tr><td>\a</td><td>bell (7)</td></tr><tr><td>\b</td><td>backspace (8)</td></tr><tr><td>\t</td><td>tab (0x09)</td></tr><tr><td>\n</td><td>newline (10)</td></tr><tr><td>\v</td><td>vertical tab (11)</td></tr><tr><td>\f</td><td>formfeed (12)</td></tr><tr><td>\r</td><td>return (13)</td></tr><tr><td>\xnn</td><td>hex value, where <i>nn</i> is a two-digit hexadecimal number</td></tr><tr><td>\nnn</td><td>decimal value, where <i>nnn</i> is a three-digit decimal number</td></tr></tbody></table>	\a	bell (7)	\b	backspace (8)	\t	tab (0x09)	\n	newline (10)	\v	vertical tab (11)	\f	formfeed (12)	\r	return (13)	\xnn	hex value, where <i>nn</i> is a two-digit hexadecimal number	\nnn	decimal value, where <i>nnn</i> is a three-digit decimal number
\a	bell (7)																		
\b	backspace (8)																		
\t	tab (0x09)																		
\n	newline (10)																		
\v	vertical tab (11)																		
\f	formfeed (12)																		
\r	return (13)																		
\xnn	hex value, where <i>nn</i> is a two-digit hexadecimal number																		
\nnn	decimal value, where <i>nnn</i> is a three-digit decimal number																		

The following example indicates a string composed only of upper- and lower-case characters, digits, periods and commas:

```
User-Name = A-Za-z0-9.,
```

The following example permits upper- and lower-case characters only:

```
User-Name = A-Za-z0-9
```

redirect.ini File

Account redirection allows you to flag an account for special processing after a configurable number of failed login attempts within a configurable period. The `redirect.ini` initialization file specifies the settings used for account redirection when users forget or mis-enter their passwords.

redirect.ini [Settings] Section

The `[Settings]` section of `redirect.ini` enables and configures account redirection settings. The following table describes the fields in the `[Settings]` section of `redirect.ini`.

Field	Meaning
Enable	If set to 0 (the default), account redirection is disabled. If set to 1, account redirection is enabled. <i>Note: Do not enable account redirection if account lockout is enabled.</i>
Lockout	The number of seconds in the account redirection lockout period. For example, a lockout period of 86,400 seconds locks a user out for one day if account redirection processing fails to authenticate the user.
Profile	The name of the global profile that supplies the values and attributes used for the user after account redirection is triggered.
Redirect	The number of seconds during which a user is in redirect state. If the redirection period elapses without another user authentication request, the user is returned to normal state.
Rejects	The number of rejected attempts prior to lockout.
Within	The period in seconds during which a specified number of rejects causes account redirection.

For example, if the `[Settings]` section of `redirect.ini` contains the following settings:

```
[Settings]
Enable = 0
Rejects = 3
Within = 180
Redirect = 120
Profile = RedirectProfile
Lockout = 86400
```

then if a user fails authentication three times within 180 seconds, the user account is placed into redirect state.

- If the user does not submit another authentication request within 120 seconds of entering redirect state, the user account is restored to normal state.

- If the user submits another authentication request within 120 seconds of entering redirect state, the user is given is accepted without authentication/authorization processing, the user's account is placed into accept-pending state, and the RADIUS accept message for the user contains the values and attributes specified in the global RedirectProfile profile. (These values or attributes could be used by an external customer process to direct the user to a secure web page that asks for alternative authentication information or billing information; the external process might then mail the user an access password if the user satisfies the external process requirements.)

When a user is in accept-pending state, the user's next authentication request determines whether Steel-Belted Radius accepts or locks out the user:

- If the next authentication is successful, the user account is returned to normal state.
- If the next authentication fails to accept the user, the user account is locked out for 86,400 seconds (one day). During this lockout period, authentication requests for this user are rejected automatically, even if the user enters the correct password.

redirect.ini [ClientExclusionList] Section

The [ClientExclusionList] section of redirect.ini identifies the RADIUS clients that are excluded from account redirection processing. Each entry in the [ClientExclusionList] section of redirect.ini consists of the name of a RADIUS client, as configured in the Steel-Belted Radius database.

RADIUS client names are case-sensitive.

spi.ini File

The spi.ini initialization file defines encryption keys and identifies the servers from which Steel-Belted Radius processes encrypted Class attributes in accounting requests. The spi.ini file allows one Steel-Belted Radius server to decode accounting requests for sessions that were authenticated on a different Steel-Belted Radius server. Class attributes received from servers not specified in spi.ini are ignored.

All Steel-Belted Radius servers that may receive authentication and accounting requests from a common NAS or AP must be configured with similar spi.ini files, which must list the IP addresses of all the servers in that "cluster." This allows one server to authenticate a user and generate an encrypted Class attribute that can be decrypted and processed by any other server in the cluster.

spi.ini [Keys] Section

The [Keys] section of spi.ini specifies the list of encryption keys used to encode subattributes encapsulated within Class attributes.

```
[Keys]
CurrentKey = n
1 = value
2 = value
.
.
.
```

Field	Meaning
CurrentKey	Specifies the encryption key that is currently active, where n is 0 or the number of a key listed in the [Keys] section: 0 – Generate and use a unique random key to encrypt Class attributes. Used only when the Steel-Belted Radius server does not exchange encrypted Class attributes with other servers. n – Use the key specified below to encrypt Class attributes.
1 = value	Specifies the number and value of the encryption key.

In the following example, the Steel-Belted Radius server generates a unique random key to encrypt Class attributes.

```
[Keys]
CurrentKey = 0
```

In the following example, the second key (*swordfish*) is currently active and used to encrypt Class attributes. The other keys in this section can be used to decrypt Class attributes received from other servers in the same cluster.

```
[Keys]
CurrentKey = 2
1 = firstkey
2 = swordfish
3 = mypassword
```

spi.ini [Hosts] Section

The [Hosts] section of spi.ini identifies the IP address of servers from which Class attributes are parsed for encapsulated/encrypted subattributes. Class attributes from servers not identified in the [Hosts] section of spi.ini are passed without special processing.

The information in the [Hosts] section may also be used to compute the server's identifier, which is included in the Class attribute. If one of a host's interfaces is included in the [Hosts] section, that interface is used to compute the server identifier. If more than one interface for a host is listed, the IP address of the last interface listed is used. If no matching address is found, the host's primary IP address is used. Addresses not corresponding to a host interface are used to configure the collection of servers whose Class attributes are accepted.

In the following example, three servers are identified as belonging to a cluster.

```
[Hosts]
192.168.15.21
192.168.23.121
192.168.23.205
```

tacplus.ini File

The tacplus.ini initialization file provides the configuration information that enables the Steel-Belted Radius server to communicate with a TACACS+ server.

tacplus.ini [ServerInfo] Section

The [ServerInfo] section of tacplus.ini provides information that allows the TACACS+ server and Steel-Belted Radius to communicate. The following fields may be present:

tacplus.ini	
[ServerInfo] Field	Meaning
SharedSecret	The shared secret between the TACACS+ server and Steel-Belted Radius.
TargetHost	The IP address of the TACACS+ server.

For example:

```
[ServerInfo]
SharedSecret=123abc
TargetHost=197.43.160.101
```

update.ini File

The update.ini initialization file controls what information is updated when Steel-Belted Radius receives a HUP or USR2 signals, which are sent via the UNIX signal command on Solaris and via the radhup.exe and radusr2.exe programs on Windows.

When Steel-Belted Radius receives a HUP or USR2 signal, it performs the tasks specified in the [HUP] and [USR2] sections of the update.ini file. You can perform tasks selectively by modifying update.ini to toggle specific settings, issuing a HUP signal to initiate enabled one set of tasks, and then modifying update.ini and issuing another HUP signal to initiate a different set of tasks.

update.ini [HUP] and [USR2] Sections

The [HUP] section of update.ini specifies what tasks Steel-Belted Radius should perform when it receives a HUP signal. The [USR2] section of update.ini specifies what tasks Steel-Belted Radius should perform when it receives a USR2 signal.

The following fields may be present in the [HUP] or [USR2] section of update.ini. If a field is not present, it defaults to 0 (disabled).

update.ini	
[HUP] or [USR2]Field	Meaning
ResetStats	If set to 0, do not reset Steel-Belted Radius statistics to 0 when a HUP or USR2 signal is received. If set to 1, reset Steel-Belted Radius statistics to 0 when a HUP or USR2 signal is received.
Update3GPP2	If set to 0, do not update 3GPP2 settings from 3gpp2.ini when a HUP or USR2 signal is received. If set to 1, update 3GPP2 settings from 3gpp2.ini when a HUP or USR2 signal is received. <i>Note: this setting only applies if Steel-Belted Radius is running on a server on which 3GPP2 is licensed and enabled.</i>

update.ini [HUP] or [USR2]Field	Meaning
UpdateAutoStop	<p>If set to 0, do not update the Proxy AutoStop settings (by re-reading the AcctAutoStopEnable setting in radius.ini) when a HUP or USR2 signal is received.</p> <p>If set to 1, update the Proxy AutoStop settings (by re-reading the AcctAutoStopEnable setting in radius.ini) when a HUP or USR2 signal is received.</p> <p><i>Note: When Proxy AutoStop is enabled, an AutoStop request is automatically recorded and associated with the session in the current sessions database when an Accounting-Start message is received.</i></p>
UpdateCCAGateways	<p>If set to 0, do not update 3Com CCA gateways (specified in ccagw.ini) when a HUP or USR2 signal is received.</p> <p>If set to 1, update 3Com CCA gateways (specified in ccagw.ini) when a HUP or USR2 signal is received.</p>
UpdateConcurrency	<p>If set to 0, do not update Concurrency Server settings when a HUP or USR2 signal is received.</p> <p>If set to 1, update Concurrency Server when a HUP or USR2 signal is received.</p> <p><i>Note: this setting only applies if Steel-Belted Radius is running on a Concurrency Server.</i></p>
UpdateDHCPPools	<p>If set to 0, do not update DHCP pool settings specified in dhcp.ini when a HUP or USR2 signal is received.</p> <p>If set to 1, update DHCP pool settings specified in dhcp.ini when a HUP or USR2 signal is received.</p>
UpdateLogAndTraceLevel	<p>If set to 0, do not update log and trace levels specified in radius.ini when a HUP or USR2 signal is received.</p> <p>If set to 1, update log and trace levels specified in radius.ini when a HUP or USR2 signal is received.</p>
UpdatePAS	<p>If set to 0, do not update PAS Server settings when a HUP or USR2 signal is received.</p> <p>If set to 1, update PAS Server settings when a HUP or USR2 signal is received.</p> <p><i>Note: this setting only applies if Steel-Belted Radius is running on a Port Allocation System (PAS) Server.</i></p>
UpdatePlugins	<p>If set to 0, do not update plug-ins that support dynamic re-reading of configuration settings when a HUP or USR2 signal is received.</p> <p>If set to 1, update plug-ins that support dynamic re-reading of configuration settings when a HUP or USR2 signal is received.</p> <p><i>Note: The TLS, TTLS, and PEAP plug-ins currently support dynamic configuration updates.</i></p>

update.ini	
[HUP] or [USR2]Field	Meaning
UpdateProxy	If set to 0, do not update realm configuration when a HUP or USR2 signal is received. If set to 1, update realm configuration (by re-reading proxy.ini, *.pro, and *.dir files) when a HUP or USR2 signal is received.
UpdateValuePools	If set to 0, do not update attribute value pool settings (in *.rr files) when a HUP or USR2 signal is received. If set to 1, update attribute value pool settings (in *.rr files) when a HUP or USR2 signal is received.

Sample update.ini File

The update.ini file installed with Steel-Belted Radius is presented below. This file causes Steel-Belted Radius to re-read all settings when it receives a HUP signal and to clear its statistics when it receives a USR2 signal.

```
[HUP]
UpdateLogAndTraceLevel = 1
UpdateProxy = 1
UpdateDHCPPools = 1
UpdateCCAGateways = 1
UpdateConcurrency = 1
UpdatePAS = 1
Update3GPP2 = 1
UpdateAutoStop = 1
UpdateValuePools = 1
UpdatePlugins = 1
ResetStats = 0

[USR2]
UpdateLogAndTraceLevel = 0
UpdateProxy = 0
UpdateDHCPPools = 0
UpdateCCAGateways = 0
UpdateConcurrency = 0
UpdatePAS = 0
Update3GPP2 = 0
UpdateAutoStop = 0
UpdateValuePools = 0
UpdatePlugins = 0
ResetStats = 1
```

vendor.ini File

The vendor.ini initialization file contains information that allows Steel-Belted Radius to work with the products of other vendors.

vendor.ini [Vendor-Product Identification] Section

The [Vendor-Product Identification] section of vendor.ini identifies and provides information about the network access servers that can be used with Steel-Belted Radius. For each make/model of vendor product, the following fields may be present:

vendor.ini	
[Vendor-Product Identification] Field	
Field	Meaning
Vendor-Product	This required field specifies the name of the product. A product name must be unique, cannot include blanks and must consist of 31 or fewer characters. These product names are used only in the Make/model pull-down list in the RAS Clients dialog. This list is used when adding a new client or when selecting a vendor-specific attribute.
Dictionary	This required field specifies the dictionary file to use for this product. The dictionary file must be located in the same directory as the Steel-Belted Radius daemon or service (usually C:\RADIUS\service). You do not need to specify an extension on the dictionary name; Steel-Belted Radius automatically attaches an extension of .DCT to the dictionary names listed in this field. See "Dictionary Files" on page 239.
Send-Class-Attribute	If set to No , the Class attribute is not sent to the client on Access-Accept. (This feature is designed to accommodate devices that don't handle this attribute properly.) The default is Yes .
Send-Session-Timeout-on-Challenge	If set to Yes , the Session-Timeout attribute is sent to the client on Access-Challenge responses that include EAP messages. This attribute advises a NAS on how long it should wait for a user response to the challenge. The default is No .
Ignore-Acct-Ss	If set to Yes , the digital signature of accounting packets based on the shared secret is ignored. This is to accommodate devices that don't properly sign accounting packages. The default is No .

vendor.ini	
[Vendor-Product Identification] Field	
Field	Meaning
Ignore-Ports	<p>This field determines whether Steel-Belted Radius may infer that one user has logged off if the port that was in use is now being used by another user.</p> <p>If set to <code>No</code>, then such an inference is made and the previous user is removed from the Active Users list. If set to <code>Yes</code>, then no such inference is made and both users are deemed active.</p> <p>The default is <code>No</code>.</p>
Discard-After	<p>Used for inbound Proxy RADIUS servers that send username information in a “decorated” format. For example, if a Proxy RADIUS server sends usernames of the form <code>username@company</code>, then specifying <code>'@'</code> results in all text after the <code>@</code> delimiter character being discarded for authentication purposes; the string <code>username</code> is used.</p>
Discard-Before	<p>Same as discard-after, except the name is on the right of the delimiter character and discardable information is on the left.</p>
Max-EAP-Fragment	<p>You can specify the size of the maximum EAP-Message in the <code>tlsauth.aut</code> and <code>tlsauth.eap</code> files. The maximum fragment length defaults to 1020. This is inefficient, however, as the fragment length must be set to a number low enough to work with all of a customer's Access Points.</p> <p>This setting allows specifying a maximum EAP fragment length on a make/model basis. The maximum EAP fragment length emitted by TLS or TTLS is the lesser of the maximum specified in their <code>.eap/.aut</code> files and this setting.</p>

vendor.ini Product-Scan Settings

After you define a Vendor-Product entry in `vendor.ini`, the name of this entry can be selected in the RAS Clients dialog as a possible value for the Make/model field. The Product-Scan-Auth and Product-Scan-Acct settings can be used within a Vendor-Product entry to permit dynamic make/model selection to occur. That is, these settings enable Steel-Belted Radius to examine the incoming packet to determine the make/model of the NAS device that originated the packet.

A dynamic Vendor-Product entry might appear as follows:

```
Vendor-Product = DeviceNameInRASClientsList
Product-Scan-Auth = MakeModelSelect
Product-Scan-Acct = MakeModelForAccounting
```

```
[MakeModelForAuthentication]
Product = String
```



```

Product = String
.
.
.
Product =

[MakeModelForAccounting]
Product = String
Product = String
.
.
.
Product =

```

The meaning of these fields is as follows:

vendor.ini

Product-Scan Field

Meaning

Vendor-Product	This setting creates a label that appears as a selection in the Steel-Belted Radius Administrator program, RAS Clients dialog, Make/model drop-down list.
Product-Scan-Auth= <i>name</i>	This field applies only to authentication servers. <i>name</i> references a section heading that appears elsewhere in vendor.ini.
Product-Scan-Acct= <i>name</i>	This field applies only to accounting servers. <i>name</i> references a section heading that appears elsewhere in vendor.ini.
[<i>name</i>]	Provides rules that govern dynamic make/model selection. These rules apply on an authentication server if the value <i>name</i> is assigned to Product-Scan-Auth; they apply on an accounting server if the value <i>name</i> is assigned to Product-Scan-Acct.

vendor.ini

Product-Scan Field

Meaning

Product=String

Product is a product name. *String* is a regular expression to match against attributes in the packet. Character by character, *Product* must exactly match a Vendor-Product value defined elsewhere in the vendor.ini file.

Product=

The default vendor.ini that is provided with Steel-Belted Radius comes equipped with a large number of Vendor-Product values from which you may choose. Each corresponds to a vendor-specific RADIUS attribute dictionary that is also provided.

The list of product names and strings is tried in order. If the packet does not come from the first device, the next is tried, and so on until the last entry in the list is tried.

You can set up a default at the end of the list by making sure the last Product entry in the list has no String assigned. If no match is found earlier in the list, Steel-Belted Radius assumes that the packet comes from the type of device.

The following example would be appropriate in a configuration whose NASs were mostly Ascend devices:

```
Product-Scan-Auth = Bigco Special Scan
.
.
.
[Bigco Special Scan]
Ascend MAX Family = \x2c?
Nortel Versalar Remote Access Concentrator =
    \x1a?\x00\x00\x06\x30
US Robotics NETServer = \x1a?\x00\x00\x01\xad
Ascend MAX Family =
```

The above example sets up dynamic make/model selection for authentication and states that the identity of the client device should be determined by seeking matches in the following order:

- 1 Is the attribute with identifier number 0x2c (Acct-Session-Id), with a value of any length (indicated by the question mark character), found in the incoming authentication packet? If so, the originating NAS is a member of the Ascend MAX Family; use that vendor-specific dictionary.
- 2 Is the vendor-specific attribute with identifier number 0x1a (Vendor-Id), with a value of any length (indicated by the question mark character), present in the packet? If so, does it have the value 1584 (0x630) which indicates a

Nortel Networks Versalar RAC? If so, use that vendor-specific dictionary (provided with Steel-Belted Radius).

- 3 Is the `Vendor-Id` attribute present, with any length, and if so, does it have the value 429 (0x1ad) which indicates a US Robotics NETServer? If so, use that vendor-specific dictionary (provided with Steel-Belted Radius).
- 4 If no match can be found using the rules specified in this section, then use the vendor-specific dictionary for the Ascend MAX Family.

Note: You should include a default entry in this section. When there is no default, if an Access-Request is received with no vendor-specific attributes of any kind, the user may be rejected due to invalid resources, as the RADIUS server cannot associate a valid dictionary with the request. Using the example:

*- Standard Radius - =
as the last line in this section is a safe configuration.*

Dictionary Files

For each product listed in the `vendor.ini` file, Steel-Belted Radius provides a dictionary file (`.dct`). Dictionary files enable Steel-Belted Radius to exchange attributes with RADIUS clients of that product type. A dictionary file provides the information that the server needs:

- When receiving RADIUS requests, to know which attributes it should expect to receive from devices of a certain product type.
- When composing a RADIUS response, to include the specific reply attributes required by devices of that product type.

Windows

Dictionary files must be placed in the same directory as the Steel-Belted Radius service (usually `C:\RADIUS\Service`). While starting up, Steel-Belted Radius scans its home directory for all files with an extension of `.dct` (regular dictionary files) or `.dci` (import dictionary files) and concatenates them into a single dictionary.

UNIX

Dictionary files must be placed in the same directory as the Steel-Belted Radius daemon. During initialization, Steel-Belted Radius reads the file `dictiona.dcm` in the

server directory to get a list of files with an extension of .dct (regular dictionary files) or .dci (import dictionary files) and concatenates them into a single dictionary.

Dictionary File Records

Records in a dictionary file must begin with one of the following keywords. Each keyword is described in detail below.

Keyword	Meaning
@	Include the referenced file
ATTRIBUTE	Define a new attribute
VALUE	Define a named integer value for an attribute
MACRO	Define a macro used to simplify repetitive definitions
OPTIONS	Define options beyond the scope of attribute definitions
#	Ignore this text (comment)

Editing Dictionary Files

The product-specific files shipped with Steel-Belted Radius reflect specific vendors' implementations of RADIUS clients. Therefore, you do not usually need to modify the dictionary files shipped with Steel-Belted Radius. However, if you are in communication with your NAS vendor about a new product, a new attribute, or a new value for an attribute, you can add this information to your existing Steel-Belted Radius configuration by editing dictionary files.

Before you edit an existing dictionary file or create a new one, you must do the following to integrate your changes into Steel-Belted Radius:

- 1 Add a new vendor-product entry to vendor.ini so that you can reference the new dictionary while configuring Steel-Belted Radius.
See “vendor.ini File” on page 235.
- 2 Place your dictionary file in the same directory as the Steel-Belted Radius service (usually C:\RADIUS\Service) or daemon.
- 3 Edit the dictiona.dcm file so that it includes your new dictionary file.
- 4 Stop and restart the server.

Include Records

Records in a dictionary file that begin with the '@' character are treated as special include records. The string that immediately follows the '@' character identifies the

name of a dictionary file whose contents are to be included. For example, the entry `@vendorA.dct` would include all of the entries in the file `vendorA.dct`.

Include records are honored only one level deep. If, for example, file `vendorA.dct` specifies an inclusion of file `radbase.dct` that, in turn, includes `radacct.dct`, `vendorA.dct` are considered to include all records in `radbase.dct`, but not those in `radacct.dct`.

Master Dictionary File

The master dictionary `dictiona.dcm` consists of include records that reference the various vendor-specific dictionaries. The order in which the vendor-specific dictionaries are included in the master dictionary has significance only if there are two vendor-specific dictionaries that contain conflicting definitions for the same attribute or attribute value.

As with standard dictionary file processing, the earlier definition of the attribute or attribute value takes precedence over any later definitions of the same attribute or attribute value. For example:

```
@vendorA.dct
@vendorB.dct
@vendorC.dct
@vendorD.dct
```

One limitation of standard dictionary files (that the `attrib_id` of all the attribute records must be unique) is waived for the master dictionary file. Multiple vendors may well define different attribute names for the same attribute identifier (assuming the attribute identifier is not already used in the base RADIUS specification). Since attributes in the Steel-Belted Radius database are stored by name (rather than by `attrib_id`), the waiver of this rule introduces no ambiguity into the database.

Import Dictionary Files

Import dictionary files (`.dci`) are special dictionary files that are used only when importing user data from the users text file supported by UNIX implementations of RADIUS.

See “Importing from Other File Formats” on page 141.

The purpose of the import dictionary files is to map the names of attributes commonly used in these implementations to those that are in use in Steel-Belted Radius.

When the Administrator program is used to import user data from an external text file, the pull-down that requests information about the type of the text file contains the list of all import dictionary files for which a standard dictionary file by the same

name is present. Importing the user data from the text file causes the import dictionary to be used to translate the attribute names to attribute IDs and the standard dictionary to convert the attribute IDs back to attribute names.

The expected format of records in the import dictionary files is identical to that of records in standard dictionaries.

ATTRIBUTE Records

Attribute records define new attributes and conform to the following syntax:

```
ATTRIBUTE attrib_name attrib_id syntax_type flags
```

where the parameters have meaning as follows:

Parameter	Meaning
<code>attrib_name</code>	Name of the attribute (up to 31 characters with no embedded blanks)
<code>attrib_id</code>	Integer in the range 0 to 255 identifying the attribute's encoded identifier
<code>syntax_type</code>	Syntax type of the attribute.
<code>flags</code>	Defines whether an attribute appears in the Check-List, the Return-List (or both), whether it is multi-valued and whether it is orderable.

The following example illustrates a typical attribute record:

```
ATTRIBUTE Framed-IP-Netmask 9 ipaddr Cr
```

This attribute record specifies all of the following:

- An attribute named Framed-IP-Netmask is supported.
- Its encoded identifier is 9.
- It must use the syntax of an IP address.
- It can appear multiple times in a Check-List and at most one time in a Return-List for User or profile entries in the Steel-Belted Radius database.

Attribute Name and Identifier

No two attribute records in a single dictionary file should have the same `attrib_name` or `attrib_id`. If a duplicate `attrib_name` or `attrib_id` is encountered, the later definition of the attribute is ignored in favor of the earlier one (the earlier one is considered to be an override).

Syntax Type Identifier

The supported standard `syntax_type` identifiers are:

Syntax Type	Meaning
hexadecimal	Hexadecimal string
hex1, hex2, hex4	1-, 2- or 4-byte hexadecimal number
int1, int2, int4, integer	1-, 2- or 4-byte decimal number (integer is equivalent to int4)
ipaddr	IP address or IP netmask attribute
ipaddr-pool	IP address selected from an IP address pool
ipxaddr-pool	IPX network number selected from an IPX address pool
string	String attribute (includes null terminator)
stringnz	String attribute (without null terminator)
time	Time attribute (number of seconds since 00:00:00 GMT, 1/1/1970)

Compound Syntax Types

In addition to the standard `syntax_type` identifiers listed above, the dictionary can accommodate compound syntax types for use in defining vendor-specific attributes. Instead of a single `syntax_type` identifier, one or more of the following options can be combined inside square brackets to form a compound syntax type:

Option	Meaning
<code>vid=nnn</code>	The device manufacturer's SMI Network Management Private Enterprise code (assigned by ISO) in decimal form.
<code>typeN=nnn</code>	Type field for vendor-specific attribute as defined in the RADIUS specification; <i>N</i> specifies the length of the field (in bytes), <i>nnn</i> specifies the decimal value of the field.
<code>lenN=nnn</code>	Length field for vendor-specific attribute as defined in the RADIUS specification; <i>N</i> specifies the length of the field (in bytes), <i>nnn</i> specifies the decimal value of the field (a plus sign prior to the value indicates that the length of the data portion is to be added to <i>nnn</i> to obtain the actual length).
<code>data=syntax_type</code>	The actual data to be included in the attribute; the syntax can be any of the standard syntax types.
<code>tag=nnn</code>	Tunnel attributes include a tag field, which may be used to group attributes in the same packet which refer to the same tunnel. Since some vendors' equipment does not support tags, this syntax type is optional and must be present in order for the attribute to include a tag field. A value of 0 indicates that the field should be present but ignored.

An example of a vendor-specific attribute definition follows:

```
ATTRIBUTE vsa-xxx 26 [vid=1234 type1=1 len1=+2 data=string] R
```

Flag Characters

The `flags` field consists of the concatenation of one or more characters from the following list:

Flag Character	Meaning
b or B	Indicates that an attribute may be bundled in a single Vendor-Specific-Attribute for a particular vendor id. That is, it may be included as one of a series of subattributes within a single VSA.
c	Attribute can appear a single time within a user or profile check-list.
C	Attribute can appear multiple times within a user or profile check-list.
r	Attribute can appear a single time within a user or profile return-list.
R	Attribute can appear multiple times within a user or profile return-list.
t	Attribute can appear a single time within a tunnel attribute list.
T	Attribute can appear multiple times within a tunnel attribute list.
o or O	Attribute is orderable; the administrator can control the order in which such attributes are stored in the Steel-Belted Radius database (this flag makes sense only for multi-valued attributes).

VALUE Records

Value records are used to define names for specific integer values of previously defined integer attributes. Value records are never required, but are appropriate where specific meaning can be attached to an integer value of an attribute. The value record must conform to the following syntax:

```
VALUE attrib_name value_name integer_value
```

where the parameters have meaning as follows:

Parameter	Meaning
<code>attrib_name</code>	Name of the attribute (up to 31 characters with no embedded blanks)
<code>value_name</code>	Name of the attribute value (up to 31 characters with no embedded blanks)
<code>integer_value</code>	Integer value associated with the attribute value

No two value records in a dictionary file should have the same `attrib_name` and `value_name` or the same `attrib_name` and `integer_value`. If a duplicate is encountered, the later definition of the attribute value is ignored in favor of the earlier one (the earlier one is considered to be an override).

The following example illustrates the use of the VALUE record to define more user-friendly attribute values for the Framed-Protocol attribute:

```
ATTRIBUTE Framed-Protocol 7 integer Cr
```


VALUE	Framed-Protocol	PPP	1
VALUE	Framed-Protocol	SLIP	2

Using these dictionary records, the administrator need not remember that the integer value 1 means PPP and the integer value 2 means SLIP when used in conjunction with the Framed-Protocol attribute. Instead, the Steel-Belted Radius Administrator program lets you choose from a list of attribute values including PPP and SLIP.

MACRO Records

Macro records are used to streamline the creation of multiple vendor-specific attributes that include many common parameters. A macro record can be used to encapsulate the common parts of the record. The macro record must conform to the following syntax:

```
MACRO macro_name(macro_vars) subst_string
```

where the parameters have meaning as follows:

Parameter	Meaning
<code>macro_name</code>	Name of the macro
<code>macro_vars</code>	One or more comma-delimited macro variable names
<code>subst_string</code>	String into which macro variables are to be substituted; any sequence of characters conforming to the format <code>%x%</code> for which a macro variable called <code>x</code> has been defined undergo the substitution process

The following example illustrates the use of a macro that simplifies the specification of multiple vendor-specific attributes:

```
MACRO Cisco-VSA(t, s) 26 [vid=9 type1=%t% len1=+2 data=%s%]
ATTRIBUTE Cisco-xxx Cisco-VSA(1, string) R
ATTRIBUTE Cisco-yyy Cisco-VSA(4, int4) C
ATTRIBUTE Cisco-zzz Cisco-VSA(9, ipaddr) r
```

The macro preprocessor built into the Steel-Belted Radius dictionary processing would translate the records in the example above to the following records before being processed.

```
ATTRIBUTE Cisco-xxx 26 [vid=9 type1=1 len1=+2 data=string] R
ATTRIBUTE Cisco-yyy 26 [vid=9 type1=4 len1=+2 data=int4] C
ATTRIBUTE Cisco-zzz 26 [vid=9 type1=9 len1=+2 data=ipaddr] r
```

OPTION Records

By default, each vendor-specific attribute is encoded in a single VSA record. The format of a VSA record is as follows:

Bits	Field
0 - 7	Type: contains the value 26.
8 - 16	Length of data in bytes.
17 - 47	Vendor ID
48 - on	Vendor data

If you provide a parameter to the `OPTION` setting, however, multiple vendor-specific attributes can be present in the vendor-data portion of a single VSA record.

The `OPTION` record must conform to the following format:

```
OPTION bundle-vendor-id = vid
```

Important: *You must also set the `B` flag in order for attribute bundling to happen. That is, in order for a particular vendor-specific attribute to be bundled, you must both set the `OPTION` record for the vendor's vendor-ID and set the `B` (or `b`) flag for the specific attribute.*

The Nortel Rapport dictionary supports this option, for example. If you want to combine Nortel's vendor-specific attributes in a single VSA, you would provide the entry:

```
OPTION bundle-vendor-id=562
```

This is because 562 is Nortel's Vendor ID, as set in the `MACRO` record. The Nortel Rapport vendor-specific attributes now would be concatenated within the vendor-data portion of a `RADIUS` VSA attribute (up to 249 octets).

services File

Steel-Belted Radius reads the `services` file at startup. Among the items of information in the `services` file are the port assignments for `RADIUS` authentication and accounting services. The location of the file depends on your operating system:

- Under **UNIX**: `/etc/` (may also be mapped using `NIS` or `NIS+`)
- Under **Windows**: `C:\winnt\system32\drivers\etc\`

The Steel-Belted Radius server uses the following default UDP ports:

- 1645 for RADIUS authentication
- 1646 for RADIUS accounting

The Steel-Belted Radius server can be configured to use any available UDP ports for authentication and accounting. You can configure new default assignments for these ports as follows:

- 1 Open the `services` file using any text editor.
- 2 To set the port for authentication, set the value of the `radius` parameter.
- 3 To set the port for accounting, set the value of the `radacct` parameter. For example:

```
radius 1812/udp # entry for radius authentication
radacct 1813/udp # entry for radius accounting
```

If there is no entry in the `services` file for `radius` or `radacct`, the Steel-Belted Radius server uses the default values (1645 and 1646, and 1812 and 1813).

Note: Any assignments made in the `radius.ini` file override the assignments made in this file.

See “radius.ini [Ports] Section” on page 220 and “RADIUS Ports” on page 35.

You can determine the ports that Steel-Belted Radius is using at any time by examining the Radius log files `yyyymmdd.LOG` or the Accounting log files `yyyymmdd.ACT` for that time period.

Attribute Value Pools (*.rr files)

This advanced feature of Steel-Belted Radius assigns attribute sets dynamically when an Authorization Request is processed and returns them in an Access-Accept.

This functionality is supported by the use of a VSA called `Funk-Round-Robin-Group`. The value for this attribute is a string, and should be set to the name of a `.rr` suffix file that defines an Attribute Value Pool. This value can therefore be set for a User or profile by using the Administrator dialog, by any other return-list mechanism (such as database retrieval).

A `.rr` file is defined as follows:

```
[Sets]
SetName1 = Weight1
SetName2 = Weight2
...
[SetName1]
```

```
AttributeNamel.1 = AttributeValue1.1
AttributeNamel.2 = AttributeValue1.2
...
```

Steel-Belted Radius maintains “round-robin” statistics for each Attribute Value Pool so that weight calculations can be performed properly. When a user logs in who belongs to a profile that has been assigned to a particular Attribute Value Pool, the round-robin values are incremented to determine which Attribute Value set should be assigned to the user. This attribute set is added to the return-list of the Access-Accept.

There are a number of scenarios that might exploit this feature. Imagine, for example, that a company wants off-site employees to be able to establish tunnels to the company network. Assume that there are three Tunnel Connection Endpoints to which end users can create VPNs into the corporate network, each of these with different capacities.

This scenario can be accommodated by defining an Attribute Value Pool of three attribute sets, each describing how to establish a tunnel with one of these connection points. These attribute sets should be weighted according to the capacity of the three connection points. The .rr file might look as follows:

```
;acme.rr
[Sets]
VPN1=20
VNP2=12
VPN3=7

[VPN1]
Tunnel-Server-Endpoint = 8.4.2.1
Tunnel-Password = GoodGuess

[VPN2]
Tunnel-Server-Endpoint = 8.4.2.2
Tunnel-Password = BestGuess

[VPN3]
Tunnel-Server-Endpoint = 8.4.2.4
Tunnel-Password = OurSecret
```

To make this Attribute Value Pool visible, you would need to define a Funk-Round-Robin-Group VSA and assign it to the users (or the profile assigned to these users) and make the value of the VSA point to the above .rr file.

```
Funk-Round-Robin-Group = acme.rr
```

Consider how you might combine Attribute Value Pools with other features. By specifying an IP Pool name for a `Framed-IP-Address` attribute, for example, you could load balance IP Pools.

Important: *Attribute merging rules do not apply to the attributes in round-robin files. It is up to the administrator to follow appropriate attribute usage (single-valued, multi-values, checklist, etc.) - no special checks are performed to ensure that the attributes and values specified in round-robin files are consistent with the rest of your system configuration. You should check the dictionary file if you have any doubts about correct attribute usage.*

Attribute Value Pools can be reconfigured dynamically. Depending on your platform:

- Under **UNIX**: Issue the HUP signal to the Steel-Belted Radius process.
- Under **Windows**: Run `RADHUP.EXE` from the command shell.

The modified files are re-read and the pool configuration reset appropriately.

Note: You can have only one active Round-Robin-Group attribute at any one time.

Auto-Restart Files (UNIX only)

When enabled, the auto-restart module acts as a watchdog daemon, monitoring the health of the Steel-Belted Radius server executable and restarting it as needed. Automatic restart is disabled by default.

Perl must be installed on the Steel-Belted Radius server if you want the automatic restart module to issue SNMP traps. Perl support is not required for syslog but is available.

Perl SNMP Support

Perl SNMP support resides in the `Perl SNMP_Session` module, which provides access to remote SNMP agents. Refer to the `readme` file for `radiusd` for information on how to install and configure the `Perl SNMP_Session` module.

Perl SNMP support allows Steel-Belted Radius to send SNMP traps to a variety of SNMP agents, including the Sun Management Center, which is distributed with some Sun hardware platforms. Sun Management Center is not required to run `radiusd`.

S90radius Script

To enable the auto-restart module, you must edit the `S90radius` script to ensure that a certain line in the script is uncommented (the hash mark ‘#’ is removed from the start of the line), as follows:

- 1 If Steel-Belted Radius is already running, become superuser and type the following command to stop the server:
`/etc/rc2.d/S90radius stop`
- 2 Edit the script `/etc/rc2.d/S90radius`. The line you want to edit for auto-restart appears as follows:

```
# RADIUS="$RADIUSDIR/radiusd --server $RADIUSDIR/radius"
```

The `--server` option identifies the location and name of the Steel-Belted Radius executable file, and must be present on the `radiusd` command line.
- 3 If the hash mark (#) is present at the start of the line, remove it.
- 4 Save and exit the file, then type the following command to restart the server: `S90radius` invokes `radiusd`, which in turn starts `radius`:
`/etc/rc2.d/S90radius start`

radiusd Script

If you enable the auto-restart module, the `S90radius` startup/shutdown script runs `radiusd` instead of the `radius` executable file. `radiusd` executes `radius` as a child process and monitors its health by a polling mechanism. Polling parameters are configurable by editing the `radiusd.conf` file in the server directory; the relevant timeouts and logging options are near the beginning of the file.

The default `radiusd.conf` settings cause the auto-restart feature to work as follows:

If the `radius` server executable fails to respond to status polling from `radiusd` within 17 seconds, `radiusd` attempts to stop `radius` using `SIGTERM` (a polite shutdown). If `radius` does not shut down within 60 seconds, `SIGKILL` (a hard kill) is used to stop it. After shutdown by either method, `radiusd` starts a new `radius` child process. If this `radius` child does not respond to status polling within 60 seconds of startup, it is presumed dead; a misconfiguration of the server is assumed; and `radiusd` terminates with a critical error.

Note: The `radius` executable normally runs as a daemon. When the automatic-restart module is enabled, the `radius` executable is run as a child process of `radiusd` instead of being run as a daemon.

While the auto-restart module is enabled, all informational, debugging, warning, error, and critical messages from `radiusd` are recorded in the following locations:

- **Syslog** – Messages are written to the UNIX `syslog` system logging facility.
- **Log file** – If `syslog` is not available, messages are written to the log file specified using the `--logfile` option on the `radiusd` command line in the `/etc/rc2.d/S90radius` script; for example:

```
RADIUS="$RADIUSDIR/radiusd \  
--server $RADIUSDIR/radius \  
--logfile /var/log/radd.log"
```

If the `--logfile` option is not already included in the `radiusd` command line, you may add it.

Note: Options processed by `radiusd` are preceded by two dashes (`--`). Options preceded with a single dash are passed to Steel-Belted Radius.

Note: If Perl isn't installed in the `/usr/local/bin/` directory and SNMP trap generation is enabled, the following error message occurs when you try to start the Steel-Belted Radius server:

```
./S90radius: /RadiusHome/radiusd: not found
```

To fix this error, edit the first line of the `radiusd` file in the `RADIUS` private directory so that the directory structure points to Perl:

```
#!/usr/local/bin/perl
```

Once this line points to Perl, SNMP trap generation should work.

Script Configuration

The `radiusd.conf` configuration file provides settings for the `radiusd` automatic-restart module.

radiusd.conf Parameter	Function
<code>WatchdogIntervalPing</code>	Number of seconds the automatic-restart module waits between sending status inquiries. Default is 5 seconds.
<code>WatchdogIntervalMaxPong</code>	Number of seconds the automatic-restart module waits for a reply before issuing a <code>SIGTERM</code> (shutdown) message. Default is 17 seconds.

radiusd.conf Parameter	Function
WatchdogIntervalMaxStartup	Number of seconds during which the server is expected to be able to start up. Default is 60 seconds.
WatchdogIntervalMaxShutdown	Number of seconds during which the server is expected to be able to shut down. Default is 60 seconds.
SnmpManager = <i>hostname</i> <i>community port version</i>	Identifier for an SNMP management station that should receive traps from the automatic-restart module. You can specify more than one SNMP management station. For each SNMP management station, enter the following: <ul style="list-style-type: none"> • <i>hostname</i> – IP address of the SNMP management station. • <i>community</i> – SNMP community string. • <i>port</i> – UDP port number used for SNMP trap messages. UDP port 162 is the default. • <i>version</i> – SNMP version number. Default is 1. If SnmpManager is undefined, SNMP traps may still be logged, but will not be transmitted on the network.
SnmpInterface	Identifies the IP network interface to be used to generate SNMP trap messages. You can specify interfaces by name or by IP address. If you enter any , then the first IPv4 interface the automatic-restart module finds is used. If you leave this parameter blank, generation of SNMP trap messages is disabled.
SnmpCommandTrap	Specifies how SNMP trap messages should be forwarded: <ul style="list-style-type: none"> • You can specify the pathname and filename for a module or executable whose syntax matches the SMC snmptrap utility. For example: /opt/SUNWsymon/util/bin/sparc-sun-solaris2.8/snmptrap • You can specify SNMP_Session.pm to deliver SNMP traps to the management station using the Perl modules. If you leave the parameter blank (the default), SNMP trap messages are not generated.

radiusd.conf Parameter	Function
SnmpCommandUptime	<p>Specifies how the automatic-restart module determines elapsed time for timestamps in trap messages.</p> <p>You can specify the pathname and filename for a module or executable whose syntax matches the SMC uclock utility. For example: /opt/SUNWsymon/util/bin/ sparc-sun-solaris2.8/uclock</p> <p>If you leave the parameter blank (the default), the automatic restart module calculates elapsed time relative to its own start time.</p>
SnmpEnterprise	<p>Specifies the OID prefix for enterprise-specific trap messages, which is used to select the appropriate MIB for decoding traps.</p> <p>Default is 1.3.6.1.4.1.1411.1.1.</p> <p>If you leave the parameter blank, SNMP trap messages are not generated.</p>
SnmpGenericTrapType= 6	<p>Specifies the enterprise-specific trap type, which must be 6 according to the SNMPv1 standard . Do not change this value without a specific reason.</p>
SnmpTrapWatchdogStarted	<p>Specifies the trap type for messages indicating that the automatic-restart module is started.</p> <p>Default is 113.</p> <p>Enter 0 to disable this type of trap.</p>
SnmpTrapWatchdogStopped	<p>Specifies the trap type for messages indicating that the automatic-restart module is stopped.</p> <p>Default is 114.</p> <p>Enter 0 to disable this type of trap.</p>
SnmpTrapWatchdogRadius Started	<p>Specifies the trap type for messages indicating that the RADIUS server is restarted.</p> <p>Default is 115.</p> <p>Enter 0 to disable this type of trap.</p>
SnmpTrapWatchdogRadiusTerm	<p>Specifies the trap type for messages indicating that the RADIUS server is not responding and that the automatic-restart module has sent the SIGTERM signal.</p> <p>Default is 5028.</p> <p>Enter 0 to disable this type of trap.</p>
SnmpTrapWatchdogRadiusKill	<p>Specifies the trap type for messages indicating that the RADIUS server is not responding and that the automatic-restart module has sent the KILL signal.</p> <p>Default is 5029.</p> <p>Enter 0 to disable this type of trap.</p>

radiusd.conf Parameter	Function
SnmpTrapWatchdogAborted	Specifies the trap type for messages indicating that the RADIUS server is not responding and that the automatic-restart module has given up and aborted. Default is 10051. Enter 0 to disable this type of trap.
SnmpTrapWatchdogFailedInit	Specifies the trap type for messages indicating that the automatic-restart module failed to start, which may indicate a misconfiguration issue. Default is 10052. Enter 0 to disable this type of trap.

Realm Configuration

7

- Stage One of Realm Configuration
- Configuring a Proxy RADIUS Realm
- Configuring a Directed Realm
- radius.ini Realm Settings
- proxy.ini File
- Proxyrl.ini
- Proxy RADIUS Configuration (.pro) File
- Directed Realm Configuration (.dir) File

Stage One of Realm Configuration

The following table lists the steps, in order, you should follow when configuring a realm of any type for Steel-Belted Radius. It also lists the parameters that you must edit in Steel-Belted Radius configuration files to accomplish each step.

After the table, we introduce the files involved in configuring a realm. Then we outline configuration steps for each type of realm, and provide a complete set of sample configuration files. A syntax guide completes the chapter.

Realm Configuration Task	File and Section
Determine the type of realm you'll provide: Proxy RADIUS or directed.	—
Does the customer have its own RADIUS server(s), to which you'll direct requests? If so, you'll need to set up a Proxy RADIUS realm for the customer. Does the customer need you to host its RADIUS server? If so, you'll need to set up a directed authentication and/or accounting realm for the customer.	
If you have not done so already, enable the realm feature on the Steel-Belted Radius server. You must do this for either type of realm.	radius.ini [Configuration] ExtendedProxy=1
You may also enable the attribute filtering feature for Proxy RADIUS realms.	AttributeEdit=1
If you have not done so already, define delimiter conventions for realm name parsing. The delimiter conventions that you define in proxy.ini are used for all realms defined on the Steel-Belted Radius server.	proxy.ini [Configuration] RealmSuffix= RealmPrefix=
Be sure to inform the customer of the delimiter and prefix/suffix conventions you are using for realms.	
Agree upon a realm name (or DNIS grouping) with the customer (RealmName).	—
If the realm is not defined by DNIS, keep the realm name short and simple, because end users must enter it in combination with their existing usernames (for example, User@RealmName). The realm name configured on the Steel-Belted Radius server does not need to match any names in use at the customer site. The realm name must not duplicate any other target name, realm name, or tunnel name in your Steel-Belted Radius configuration.	
If the realm is defined as a DNIS grouping, the user is matched to a realm based on the Called-Station-Id.	
Continue configuration steps given below as appropriate.	—

Realm Configuration Files

To configure realms, you must edit the following files in the Steel-Belted Radius server directory.

File Name	Purpose
radius.ini	Enable and disable realm features.
proxy.ini	Store information that applies to all realms on the server.
<i>RealmName.pro</i>	For each Proxy RADIUS realm that you want to configure on the Steel-Belted Radius server, you must create a file called <i>RealmName.pro</i> , where <i>RealmName</i> is the name of the realm, and you must register this <i>RealmName</i> by listing it in the [Realms] section of the proxy.ini file.
<i>RealmName.dir</i>	For each directed authentication and/or accounting realm that you want to configure on the Steel-Belted Radius server, you must create a file called <i>RealmName.dir</i> , where <i>RealmName</i> is the name of the realm, and you must register this <i>RealmName</i> by listing it in the [Directed] section of proxy.ini.
filter.ini	Specify filters for RADIUS attributes; these filters may be referenced from the [Auth] or [Acct] section of a <i>RealmName.pro</i> or <i>RealmName.dir</i> file.

Configuring a Proxy RADIUS Realm

A Proxy RADIUS server treats a *realm* as a destination against which it performs authentication and accounting.

The following table traces the process of configuring a new Proxy RADIUS realm for Steel-Belted Radius. It also lists the sections that you must edit in configuration files to accomplish each step. No step in this process may be omitted unless this table indicates that it is optional. Begin at step 1, and continue to the end of the table.

Step	Proxy RADIUS Configuration Task	File and Section
1	Complete the preparatory steps outlined in the previous section.	—
2	Register the <i>RealmName</i> with Steel-Belted Radius.	proxy.ini [Realms] <i>RealmName</i>
3	Create a realm configuration file.	<i>RealmName.pro</i>

Step	Proxy RADIUS Configuration Task	File and Section
4	<p>Study the customer's current (or planned) RADIUS configuration. The customer's RADIUS servers are the target servers in the new realm.</p> <p>Consider the following questions: Are authentication and accounting packets directed to different RADIUS servers? What is their need for a fast-fail policy, primary-secondary server strategy, or round-robin load balancing? Are some servers used for authentication and some for accounting? What is the IP address of each RADIUS server? What UDP port and shared secret does each server use for authentication and/or accounting?</p>	—
5	<p>Does the customer want its RADIUS servers to receive Accounting-On and Accounting-Off messages? If so, add the new realm to your static proxy accounting configuration. See "Static Proxy Accounting" on page 70.</p>	<pre> proxy.ini [StaticAcct] 7=name 8=name [name] realm=RealmName </pre>
6	<p>Use the Steel-Belted Radius Administrator program to create a Proxy entry for each target in the new realm. For authentication targets, ensure that the Include in authentication list box is unchecked.</p>	Proxy dialog
7	<p>Give the customer the IP address of the Steel-Belted Radius server as well as the UDP port and shared secret it uses for authentication and accounting. Instruct the customer that for each target in the new realm, the Steel-Belted Radius server must be added to the target's database as a RADIUS client. Presumably, someone at the customer site performs this task by running the target server's RADIUS configuration utility.</p>	—
8	<p>Enable authentication in this realm.</p>	<pre> RealmName.pro [Auth] Enable=1 </pre>
9	<p>(Optional) Indicate that any realm names and delimiters are to be stripped from the User-Name before it is sent to the target server for authentication.</p>	StripRealm=
10	<p>Specify which target servers receive authentication packets. Configure load balancing and other details of realm and target selection for authentication packets.</p> <p>This is a multi-step process: (1) In the [Auth] section of the <i>RealmName.pro</i> file, set Enable to 1 and assign a name to the TargetsSection parameter; (2) create a [name] section in the file; and (3) within this section list the targets for authentication. When listing a target, use the name you assigned to it in the Proxy dialog.</p>	<pre> TargetsSection=name . . . [name] Server= </pre>

Step	Proxy RADIUS Configuration Task	File and Section
11	<p>(Optional) Specify an attribute filter to apply to authentication requests going out to the realm from the Steel-Belted Radius server.</p> <p>This is a multi-step process: (1) In the [Auth] section of <i>RealmName.pro</i>, assign a name to the FilterOut parameter; (2) create a [name] section in the filter.ini file; and (3) within the filter.ini [name] section list the rules for editing the attributes in a RADIUS authentication request packet before forwarding the packet “out” to a Proxy RADIUS realm.</p>	<pre> RealmName.pro [Auth] FilterOut=name filter.ini [name] . . . </pre>
12	<p>(Optional) Specify an attribute filter to apply to authentication responses returning into the Steel-Belted Radius server from the realm.</p> <p>This is a multi-step process: (1) In the [Auth] section of <i>RealmName.pro</i>, assign a name to the FilterIn parameter; (2) create a [name] section in the filter.ini file; and (3) within the filter.ini [name] section list the rules for editing the attributes in an authentication response packet as it returns “in” from the Proxy RADIUS realm, before relaying the packet back to the RADIUS client.</p>	<pre> RealmName.pro [Auth] FilterIn=name filter.ini [name] . . . </pre>
13	Enable Proxy RADIUS accounting in this realm.	<pre> RealmName.pro [Acct] Enable=1 </pre>
14	(Optional) Indicate that any realm names and delimiters are to be stripped from the User-Name before it is sent to the target server for accounting.	<pre> StripRealm= </pre>
15	(Optional) Indicate that accounting attributes should be logged locally on the Steel-Belted Radius server as well as being directed to the realm.	<pre> RecordLocally= </pre>
16	<p>Specify which target servers receive accounting packets. Configure load balancing and other details of realm and target selection for accounting packets.</p> <p>This is a multi-step process: (1) In the [Acct] section of the <i>RealmName.pro</i> file, set Enable to 1 and assign a name to the TargetsSection parameter; (2) create a [name] section in the file; and (3) within this section list the targets for accounting. When listing a target, use the name you assigned to it in the Proxy dialog.</p>	<pre> TargetsSection=name . . . [name] Server= </pre>

Step	Proxy RADIUS Configuration Task	File and Section
17	(Optional) Specify an attribute filter to apply to accounting requests going out to the realm from the Steel-Belted Radius server. This is a multi-step process: (1) In the [Acct] section of <i>RealmName.pro</i> , assign a name to the FilterOut parameter; (2) create a [name] section in the filter.ini file; and (3) within the filter.ini [name] section list the rules for editing the attributes in a RADIUS accounting request packet before forwarding the packet “out” to a Proxy RADIUS realm.	<i>RealmName.pro</i> [Acct] FilterOut=name filter.ini [name] . . .
18	(Optional) Specify an attribute filter to apply to accounting responses returning into the Steel-Belted Radius server from the realm. This is a multi-step process: (1) In the [Acct] section of <i>RealmName.pro</i> , assign a name to the FilterIn parameter; (2) create a [name] section in the filter.ini file; and (3) within the filter.ini [name] section list the rules for editing the attributes in an accounting response packet as it returns “in” from the Proxy RADIUS realm, before relaying the packet back to the RADIUS client.	<i>RealmName.pro</i> [Acct] FilterIn=name filter.ini [name] . . .
19	(Optional) Provide DNIS information for this realm.	<i>RealmName.pro</i> [Called-Station-ID]
20	(Optional) Specify a proxy fast-fail policy for the realm.	[FastFail]
21	(Optional) Enable Steel-Belted Radius to map the presence or absence of certain attributes or values to this realm.	proxy.ini [AuthAttributeMap] <i>RealmName</i> [AcctAttributeMap] <i>RealmName</i>
22	It's possible to load your new realm configuration dynamically, without stopping and restarting the server. Under UNIX : Issue the HUP signal to the Steel-Belted Radius process: kill -HUP ProcessID Under Windows : Run the RADHUP.EXE program from the command shell. (RADHUP.EXE is located in the server directory that you specified at installation time, usually C:\RADIUS\Service.) Steel-Belted Radius re-reads proxy.ini, filter.ini, and all .pro files in the server directory, and resets its realm configuration accordingly. <i>NOTE: Rarely, you must edit radius.ini while configuring a realm. If you do edit radius.ini, you must stop and restart Steel-Belted Radius before your new configuration is fully loaded.</i>	—

The following topics display the complete set of Proxy RADIUS realm configuration files that might result from this configuration process: radius.ini, proxy.ini, sample1.pro, and filter.ini.

Sample radius.ini Realm Settings

The following excerpt from a radius.ini file enables the realm feature and the attribute filtering feature. These two features must be enabled for our sample Proxy RADIUS realm configuration files to work:

```
[Configuration]
ExtendedProxy=1
AttributeEdit=1
```

For syntax details, see “radius.ini [Configuration] Section” on page 212.

Sample proxy.ini File

The following complete proxy.ini file registers one Proxy RADIUS realm called sample1 and adds that realm to the list of target realms for static proxy accounting.

```
[Realms]
sample1

[StaticAcct]
7=CustAOnOff
8=CustAOnOff

[CustAOnOff]
realm=sample1
```

For syntax details, see “proxy.ini File” on page 268.

Sample Proxy RADIUS (.pro) File

The following complete file must be called sample1.pro for it to work with our sample proxy.ini file above.

```
[Auth]
Enable = 1
TargetsSection = AuthTargets
RoundRobin = 2
StripRealm = 0
RequestTimeout = 5
NumAttempts = 3
FilterOut = CustAOut
FilterIn = CustAIn
```

```

MessageAuthenticator = 0

[Acct]
Enable = 1
TargetsSection = AcctTargets
RoundRobin = 1
StripRealm = 0
RequestTimeout = 5
NumAttempts = 3
FilterOut = CustAOut
; FilterIn =
RecordLocally = 1
; Block = 1

[AuthTargets]
Bunion=1
Desktop=1

[AcctTargets]
desktop

[Called-Station-ID]
8885551212
5551234

[FastFail]
MinFailures = 3
MinSeconds = 3
ResetSeconds = 30

```

For syntax details, see “Proxy RADIUS Configuration (.pro) File” on page 278.

This example expects the Steel-Belted Radius database to contain Proxy entries with target names `Desktop` and `Bunion`. These entries are required to provide the network routing information (IP address, RADIUS shared secret, and UDP ports) that allows forwarded packets to reach the target servers at the customer site.

Sample filter.ini File

The following complete sample `filter.ini` file defines the two attribute filters referenced in our `sample1.pro` file above:

```

[CustAOut]
ALLOW
EXCLUDE NAS-IP-Addr
ADD NAS-IP-Addr 1.2.3.4

[CustAIn]

```

```
EXCLUDE
ALLOW Session-Timeout
ALLOW Idle-Timeout
ALLOW Service-Type Framed
ADD Service-Type Framed
ADD Framed-IP-Address CustAPool
```

For syntax details, see “filter.ini File” on page 202.

The CustAOut filter in this example is designed to be applied to request packets coming into the Steel-Belted Radius server that is directed out to the realm. It allows all of the attributes in the packet to go out to the realm, with the exception of the RADIUS client’s IP address. It replaces this IP address with the specific “dummy” address 1.2.3.4. This filter enhances overall security by not publishing routing information to the network when it’s not necessary to do so.

The CustAIn filter in this example is designed to be applied to response packets returning to the Steel-Belted Radius server, which are relayed, in turn, to the RADIUS client. Most attributes are excluded; however, if any timeout values are returned, they’ll be allowed through. If the Service-Type attribute is present in the response and it has the value `Framed` (a string alias for the Service-Type integer value 2), it is allowed in the packet. Steel-Belted Radius adds the Service-Type attribute to the packet if it is not already there, and assigns it the value `Framed` (that is, 2).

The CustAIn filter in this example expects the Steel-Belted Radius database to contain an IP Pool entry called CustAPool. This pool specifies the customer’s valid address ranges. If this entry is not present, the CustAIn filter fails. CustAPool is referenced in the filter’s final entry, which assigns a value to the Framed-IP-Address attribute. As shown in the example, this entry causes Steel-Belted Radius to (1) add the Framed-IP-Address attribute to the packet if it is not already there; (2) select an available address from CustAPool, and (3) assign this value to the Framed-IP-Address attribute.

Configuring a Directed Realm

There are a number of circumstances in which you must configure a *directed realm*.

The following table traces the process of configuring a directed authentication and/or accounting realm for Steel-Belted Radius, step by step. It also lists the sections that you must edit in Steel-Belted Radius configuration files to accomplish each step. No step in this process may be omitted unless this table indicates that it is optional. Begin at step 1, and continue to the end of the table.

Step	Directed Realm Configuration Task	File and Section
1	Complete the preparatory steps outlined in “Stage One of Realm Configuration” on page 256.	—
2	Register the RealmName with Steel-Belted Radius.	proxy.ini [Directed] <i>RealmName</i>
3	Create a realm configuration file.	RealmName.dir
4	Get the customer's user data and add it to your database, which may be an external database (SQL, LDAP) or the Steel-Belted Radius database. When adding a few entries, see “Users Dialog” on page 91. When adding many entries, see “Import/Export Capabilities” on page 138. See also “LDAP Configuration Interface” on page 332.	—
5	Configure the authentication method on the Steel-Belted Radius server. See “Configuring Authentication Methods” on page 38. See also “SQL Authentication” on page 364 and “External LDAP Authentication” on page 404.	—
6	Register the authentication method with the realm.	RealmName.dir [AuthMethods]
7	Enable directed authentication in the realm.	[Auth] Enable=1
8	(Optional) Indicate that any realm names and delimiters are to be stripped from the User-Name before authentication is performed.	StripRealm=
9	Understand the data that the customer uses (or plans to use) to store accounting and billing records. This indicates the accounting method(s) to use.	—
10	Configure the accounting method(s) on the Steel-Belted Radius server. See “proxy.ini [DirectedAcctMethods] Section” on page 273. You can set up unique accounting log files by copying <code>account.ini</code> from the server directory to another directory, renaming it (if desired, but keep the <code>.ini</code> extension), and editing it to record accounting attributes by each customer. Use <code>account.ini</code> file syntax. See “account.ini File” on page 178.	<code>.ini</code> files

Step	Directed Realm Configuration Task	File and Section
	You can also log to external SQL databases by copying an .acc file from the server directory to another directory, renaming it (if desired, but keep the .acc extension), and editing it to record accounting attributes by each customer. Use .acc file syntax. See "SQL Accounting" on page 388.	.acc files
11	Name each accounting method.	proxy.ini [DirectedAcctMethods]
12	Register the accounting method with the realm.	RealmName.dir [AcctMethods]
13	Enable directed accounting in the realm.	[Acct] Enable=1
14	(Optional) Indicate that any realm names and delimiters are to be stripped from the User-Name before accounting is performed.	StripRealm=
15	(Optional) Indicate that accounting attributes should be logged locally on the Steel-Belted Radius server as well as being directed to the realm.	RecordLocally=
16	(Optional) Provide DNIS information for this realm.	[Called-Station-ID]

Step	Directed Realm Configuration Task	File and Section
17	<p>Load your new configuration.</p> <p>If you've added or changed any directed accounting methods at all, you must stop and restart the server.</p> <p>If you've added or changed directed authentication methods in which external database (SQL or LDAP) authentication is used, you must stop and restart the server.</p> <p>If you've added or changed directed authentication methods in which local or pass-through (Native, UNIX, Domain, Host, SecurID, or TACACS+) authentication is used, it's possible to load your new realm configuration dynamically, without stopping and restarting the server.</p> <p>Under UNIX: Issue the HUP signal to the Steel-Belted Radius process:</p> <p style="padding-left: 40px;">kill -HUP ProcessID</p> <p>Under Windows: Run the RADHUP.EXE program from the command shell. (RADHUP.EXE is located in the server directory that you specified at installation time, usually C:\RADIUS\Service.)</p> <p>Steel-Belted Radius re-reads proxy.ini and all .dir files in the server directory, and resets its realm configuration accordingly.</p> <p><i>NOTE: Rarely, you must edit radius.ini while configuring a realm. If you do edit radius.ini, you must stop and restart the Steel-Belted Radius before your new configuration is fully loaded.</i></p>	

The following topics display the complete set of directed realm configuration files that might result from this configuration process: radius.ini, proxy.ini, and sample2.dir.

Sample radius.ini Realm Settings

The same radius.ini excerpt works for our sample directed realm as for our sample Proxy RADIUS realm. The ExtendedProxy field needs to be enabled (set to 1). The AttributeEdit field does not apply to directed realms.

```
[Configuration]
ExtendedProxy=1
```

For syntax details, see “radius.ini [Configuration] Section” on page 212.

Sample proxy.ini File

The following complete proxy.ini file expands the proxy.ini in our Proxy RADIUS realm example. It registers the Proxy RADIUS realm called sample1 and also, registers a directed authentication and/or accounting realm called sample2. It defines several directed accounting methods, including those we plan to reference from the sample2.pro realm configuration file.

```
[Realms]
sample1

[Directed]
sample2

[DirectedAcctMethods]
CustBAcctSQL = c:\radius\CustomerB\theirsq1.acc
CustCAcctAttributes = c:\radius\CustomerC\account.ini
CustCAcctSQLConfig = c:\radius\CustomerC\sqlacct.acc
CustDAcctSQLConfig3 = c:\radius\CustomerD\mysql.acc
```

For syntax details, see “proxy.ini File” on page 268.

Sample Directed Realm (.dir) File

The following complete configuration file must be called sample2.dir for it to work with our sample radius.ini and proxy.ini files, above.

```
[Auth]
Enable = 1
StripRealm = 1

[Acct]
Enable = 1
RecordLocally = 1

[AuthMethods]
Native User

[AcctMethods]
CustCAcctAttributes
CustCAcctSQLConfig

[Called-Station-Id]
8885551212
55512340
```

For syntax details, see “Directed Realm Configuration (.dir) File” on page 292.

This sample file configures both directed authentication and directed accounting. It also strips realm routing information from the User-Name prior to authentication.

The [Acct Methods] section of this file lists the two accounting methods for the sample2 realm. These are CustCAcctAttributes, which specifies how to log attributes to an .ACT accounting log file on the local server, and CustCAcctSQLConfig, which configures accounting to an external SQL database. Both methods are configured in the [DirectedAcctMethods] section of our sample proxy.ini file, above.

radius.ini Realm Settings

The [Configuration] section of radius.ini provides two fields that you can use to enable or disable realm features for the Steel-Belted Radius server: ExtendedProxy and AttributeEdit. Both fields are enabled (set to 1) by default. You can disable either feature by setting the corresponding field to 0.

See “radius.ini [Configuration] Section” on page 212.

The [Self] section of radius.ini allows you to list all of the realm names that should make handled by this Steel-Belted Radius server, rather than being proxied to other targets.

See “radius.ini [Self] Section” on page 224.

As with all changes to radius.ini, if you edit radius.ini while configuring a realm, you must stop and restart the Steel-Belted Radius before your new realm configuration is fully loaded.

proxy.ini File

The proxy.ini file contains information that applies to all of the realms defined on the Steel-Belted Radius server. Details of each individual realm are provided in its *RealmName.pro* or *RealmName.dir* file.

After you edit proxy.ini, you must apply your changes as follows:

- If you’ve configured any Proxy RADIUS realms, it’s possible to load your new realm configuration dynamically, without stopping and restarting the server.

Depending on your operating system:

- Under **UNIX**: Issue the HUP signal to the Steel-Belted Radius process:
kill -HUP ProcessID
- Under **Windows**: Run the **RADHUP.EXE** program from the command shell. (RADHUP.EXE is located in the server directory that you specified at installation time, usually C:\RADIUS\Service.)

Steel-Belted Radius re-reads proxy.ini, filter.ini, and all *.pro and *.dir files in the server directory, and resets its realm configuration accordingly.

- If you've configured any directed realms and if you've added or changed:
 - Any directed accounting methods at all, you must stop and restart the server to load your new configuration.
 - Directed authentication methods in which external database (SQL or LDAP) authentication is used, you must stop and restart the server to load your new configuration.
 - Directed authentication methods in which local or pass-through (Native, UNIX, Domain, Host, SecurID, or TACACS+) authentication is used, it's possible to load your new realm configuration by using a HUP signal.

Note: Rarely, you must edit radius.ini while configuring a realm. If you do edit radius.ini, you must stop and restart the server before your new configuration is fully loaded.

proxy.ini [AttributeMap] Sections

The [AuthAttributeMap] and [AcctAttributeMap] sections of proxy.ini allow you to map the presence, absence, or specific value of an attribute in the incoming packet to a specific realm. This is referred to as *attribute mapping*.

An [AuthAttributeMap] or [AcctAttributeMap] section consists of one or more *RealmName* entries. Each *RealmName* must match the name of a realm configuration file (*RealmName.pro* or *RealmName.dir*) in the same directory as proxy.ini.

Note: Attribute mapping is supported by Proxy RADIUS realms and directed realms. You cannot use this feature when forwarding packets to a Proxy target that is not a member of a realm.

Each *RealmName* entry is a list of statements that can be true or false regarding the attributes in an incoming RADIUS packet; we call these statements *rules*. Rules found in [AuthAttributeMap] apply to authentication packets; rules found in [AcctAttributeMap] apply to accounting packets. In all other respects,

[AuthAttributeMap] or [AcctAttributeMap] are the same. The syntax for individual rules may vary; the following example shows all of the possible syntax variations:

```
[AuthAttributeMap]
  RealmName
    Attribute=Value
    Attribute
    ~Attribute=Value
    ~Attribute
  .
  .
  .
[AcctAttributeMap]
  .
  .
  .
```

For example:

```
[AuthAttributeMap]
CustTRealm
  Framed-Protocol=1
  Service-Type=2
CustQRealm
  Framed-Protocol=PPP
  ~Service-Type=Framed
NativeRealm
```

Each attribute mapping rule must begin with a space or tab character, followed optionally by a tilde ('~'), then the name of a standard or vendor-specific RADIUS Attribute that is in one of the Steel-Belted Radius dictionary files. If a Value is present, it is preceded by an equal sign ('='), and must specify a valid possible value for that attribute. The rule is terminated by a carriage return. Tilde ('~') indicates that the rule is satisfied only if the attribute or attribute/value pair is not present in the packet.

Each *RealmName* entry in an [AuthAttributeMap] or [AcctAttributeMap] section is examined in sequence from top to bottom. Within each *RealmName* entry, each rule is evaluated in sequence from top to bottom. The results are as follows:

- If all of the rules in a *RealmName* entry evaluate to `true`, the packet is routed to the realm called *RealmName* and the remaining entries in the attribute map are ignored.
- If any of the rules in a *RealmName* entry evaluate to `false`, this entry does not result in a mapping. Steel-Belted Radius evaluates the next entry in the map.

- If Steel-Belted Radius encounters a *RealmName* entry that contains no rules, the packet is automatically directed to that realm.

The following table explains how the various types of rules are evaluated.

Syntax Variation	Meaning of the Attribute Mapping Rule
<i>Attribute=Value</i>	<p>If the <i>Attribute</i> is present in the request packet and it has the <i>Value</i> shown, then this rule is true. If the <i>Attribute</i> is not present, or if it is present but does not have the <i>Value</i> shown, then this rule is false.</p> <p><i>NOTE: The Steel-Belted Radius dictionary file radius.dct provides string aliases for certain integer values defined in the RADIUS standard. You are free to use these strings in attribute mapping rules.</i></p>
Attribute	<p>If the <i>Attribute</i> is present in the request packet, then regardless of its value, this rule is true. If the <i>Attribute</i> is not present, then this rule is false.</p> <p><i>NOTE: You won't often use the Attribute rule without a Value, because most of the RADIUS packets coming into your configuration are going to contain the same set of RADIUS attributes, but with different Values.</i></p>
~Attribute=Value	<p>Note the tilde (~) operator. This rule is looking for a specific attribute that may have any value except the one listed. If <i>Attribute</i> is present in the request packet and it does not have the <i>Value</i> shown, then this rule is true. If <i>Attribute</i> is not present, or if it is present but does have the <i>Value</i> shown, then this rule is false.</p> <p><i>NOTE: The following is not valid syntax: Attribute=~Value</i></p>
~Attribute	<p>Note the tilde (~) operator and the absence of a Value. If <i>Attribute</i> is not present in the request packet, then this rule is true. If <i>Attribute</i> is present, then this rule is false.</p>

When setting up [AuthAttributeMap] and/or [AcctAttributeMap] rules for your own configuration, your goal should be to distinguish between the different companies whose requests you're processing. Consider how specific your rules must be to identify each customer uniquely. Is the presence of a particular attribute enough (Ascend-IP-Address), or does the attribute need to have a specific value before you can be sure of its source (NAS-IP-Addr=n.n.n.n)? Above all, you must ensure that your logic does not permit a crossing of records between customers.

If a realm destination has been identified by applying an [AuthAttributeMap] entry to the attributes in a session's authentication request, Steel-Belted Radius uses the same realm for that session's accounting requests (if the realm is enabled for accounting). Generally, this is the desired behavior for the realm. You should provide an [AcctAttributeMap] entry only if there is no [AuthAttributeMap] entry for a realm and you want to map the realm using an accounting attribute.

proxy.ini [Configuration] Section

The [Configuration] section of proxy.ini permits you to define prefix and suffix conventions for realm name parsing; that is:

```
User<SuffixDelimiter>RealmName
```

or

```
RealmName<PrefixDelimiter>User
```

You can enable both conventions if you specify a different delimiter character for each. All prefixed name decorations must use the prefix delimiter, and all suffixed name decorations must use the suffix delimiter. For example:

```
[Configuration]
RealmSuffix = #
RealmPrefix = !
```

This [Configuration] section would enable Steel-Belted Radius to correctly route incoming User-Name values such as joeuser#nas1#nas2 or nas1!nas2!joeuser to their appropriate realms.

The default suffix delimiter is the at-sign '@' (joeuser@nas1@nas2); the default prefix delimiter is the forward slash '/' (nas1/nas2/joeuser). You may substitute for these any non-null characters (like '#' and '!' in the example above). To specify the backslash '\', which normally indicates a line continuation, use '\\'.

If you set the prefix and suffix delimiter to the same character, both prefix and suffix conventions are enabled but (since suffixes are checked first) prefixes may be misinterpreted.

We strongly suggest that you choose a different delimiter character, and possibly also a different prefix/suffix name parsing convention, for Tunnels than for Proxies or realms.

See “User-Names with a Single Delimiter” on page 60.

See also “Tunnel Name Parsing” on page 136.

proxy.ini [Directed] Section

The [Directed] section of proxy.ini lists the names of all of the directed authentication and/or accounting realms on the server.

The syntax for the [Directed] section is as follows:

```
[Directed]
RealmName
RealmName
```

.
.
where *RealmName* matches the name of a *RealmName.dir* file in the same directory as *proxy.ini*. (Do not include the filename suffix *.dir* in this entry.)

Compare “*proxy.ini [Realms] Section*” on page 275.

proxy.ini [DirectedAcctMethods] Section

The [DirectedAcctMethods] section of the *proxy.ini* file lists one or more external database accounting configuration files (*.acc*) or local accounting initialization files (*.ini*) on the local server, and assigns each of these files a name by which it may be referenced in a *RealmName.dir* file.

See “Directed Realm [AcctMethods] Section” on page 296.

The syntax for the [DirectedAcctMethods] section is as follows:

```
[DirectedAcctMethods]
Description=PathAndFile
Description=PathAndFile
.
.
.
```

where *Description* is the name by which you want to reference the accounting method, and *PathAndFile* is the full pathname of an *.acc* or *.ini* file on the local server; for example:

- Under **UNIX**:
/usr/lib/extras/acctlib.acc, or
/usr/lib/extras/ouracct.ini
- Under **Windows**:
c:\radius\extras\acctlib.acc, or
c:\radius\extras\ouracct.ini

This is the file that implements the accounting method. The location of this file must not be the Steel-Belted Radius server directory.

If your *PathAndFile* identifies an:

- *.acc* file, external database accounting is performed as configured in the file. You may reference the Steel-Belted Radius SQL accounting module in the [Bootstrap] section of this *.acc* file.
- *.ini* file, you may omit the [Bootstrap] section from this file. Normal Steel-Belted Radius logging is performed, except that:

- Accounting log entries (for requests that are routed to this accounting method) are written to accounting log files (.ACT) in the specified Path, rather than in the server directory.
- Logging details (which attributes are logged, and in which order) are controlled by the [Settings] and [Attributes] sections of the .ini file listed in *PathAndFile*, rather than the account.ini file found in the server directory.

proxy.ini [Interfaces] Section

If your server has more than one interface, you may choose to assign the outgoing proxy traffic for a particular realm to a particular interface card. If so, you must follow these steps:

- List the IP addresses associated with each card in the [Addresses] section of the radius.ini file.

See “radius.ini [Addresses] Section” on page 208.

- Create an [Interfaces] section for the proxy.ini file. This should consist of a list of one or more pairs in the following format:

```
[Interfaces]
InterfaceName = IPAddress
```

where *InterfaceName* is a label you assign to the given *IPAddress*.

- Extend the existing entries in the [*name*] sections in .pro files for proxy realms with the *InterfaceName* defined in the [Interfaces] section so that they are in the following format:

```
[TargetSection]
Target=NumAttempts,InterfaceName
```

where *InterfaceName* is the name of the interface defined above in the [Interfaces] section.

For example:

```
[Targets]
Bert=3,ABCInterface
Ernie=1,XYZInterface
```

Note: The *ProxySource* setting in the [Configuration] section of *radius.ini* overrides any proxy realm settings.

proxy.ini [Processing] Section

If this section is present, it allows you to specify which routing rules are applied and the order in which they are applied. If no [Processing] section is present, routing continues in its default behavior. The syntax is as follows:

```
[Processing]
ProxyTechnique
.
.
.
```

Field	Meaning
ProxyTechnique	This can be one of four identifiers: Attribute-Mapping, DNIS, Prefix, or Suffix. Only the rules corresponding to the values listed are applied, and they are applied in the order you specify them.

For example:

```
[Processing]
Suffix
DNIS
```

specifies that only suffix delimiter and DNIS rules should be enabled, in that order.

See “Control Over Routing Methods” on page 65.

proxy.ini [Realms] Section

The [Realms] section of proxy.ini lists all of the Proxy RADIUS realms known to the server. The syntax is as follows:

```
[Realms]
RealmName
RealmName
.
.
.
```

Field	Meaning
RealmName	Each entry must match the name of a <i>RealmName</i> .pro file in the same directory as proxy.ini.

Compare “*proxy.ini [Directed] Section*” on page 272.

proxy.ini [StaticAcct] Section

Static proxy accounting allows you to send duplicate copies of certain types of accounting request to Proxy RADIUS realms (or any RADIUS-aware device), in addition to the normal routing of the original accounting request. The number of duplicates is not limited.

See “Static Proxy Accounting” on page 70.

The [StaticAcct] section of proxy.ini maps possible values of the Acct-Status-Type attribute to a list of Proxy RADIUS realms that receive statically-forwarded, duplicate copies of all accounting packets of that type.

Acct-Status-Type is a RADIUS standard attribute that identifies the type of accounting request. The Acct-Status-Type values 1, 2, 3, 7, and 8 have been assigned names and meanings as follows. Additional values for Acct-Status-Type have been defined by NAS vendors for use with their equipment; you can also use these values in the [StaticAcct] section.

Acct-Status-Type Value	Name	Meaning
1	Start	A user session has started
2	Stop	A user session has stopped, request contains final statistics
3	Interim	A user session is in progress, request contains current statistics
7	Accounting-On	The NAS has started up
8	Accounting-Off	The NAS is about to shut down

The syntax for a [StaticAcct] section is as follows:

```
[StaticAcct]
number=name
number=name
.
.
.
```

where each *number* is a possible value of the Acct-Status-Type attribute, and each *name* identifies a section called [*name*] that appears elsewhere in the proxy.ini file.

When it receives an accounting request with an Acct-Status-Type of *number*, Steel-Belted Radius uses the [StaticAcct] section to match *number* with *name*, and statically forwards a duplicate copy of the packet to all of the Proxy RADIUS realms listed in the [*name*] section.

Each [*name*] section consists of a list name in square brackets ([*name*]) followed by a list of Proxy RADIUS realms. Each of these realms must have a *RealmName.pro*

file in the same directory as proxy.ini. Directed realms do not support static proxy accounting.

The syntax for a `[name]` section is as follows:

```
[name]
realm=RealmName
realm=RealmName
.
.
.
```

The `[name]` section is used only if its `name` is mapped to a number in the `[StaticAcct]` section of the proxy.ini file.

The following excerpt from a proxy.ini file demonstrates some of the flexibility of static proxy forwarding. Copies of all session-related accounting packets (Start, Stop, and Interim) are proxy-forwarded to a realm called billing. Copies of all device-related accounting packets (Accounting-On and Accounting-Off) are proxy-forwarded, not only to billing, but also to a realm called operations.

```
[Realms]
billing
operations

[StaticAcct]
1 = SessionObserverList
2 = SessionObserverList
3 = SessionObserverList
7 = NASObserverList
8 = NASObserverList

[SessionObserverList]
realm = billing

[NASObserverList]
realm = billing
realm = operations
```

Proxyrl.ini

The proxyrl.ini file supports a feature called *Smart Static Accounting*, which allows you to specify that the accounting packets for a proxy or directed realm should be forwarded to a list of one or more proxy realms. These groups of realms can also be used for static accounting configured in proxy.ini.

This file consists of a number of sections that you name. Each section name is referenced in the `StaticAcctRealms` field in the `[Acct]` section of a `.pro` or `.dir` file. Following the section name, you can list a number of proxy realm names, in the following format:

```
[realm-list-name-1]
proxy-realm-1
proxy-realm-2
.
.
.
[realm-list-name-2]
.
.
.
```

For example:

```
[StaticAcctTargets1]
AcctSrvr1
AcctSrvr4
```

Warning: You must be sure that the list of static accounting servers doesn't include any realms that use the list or an infinite loop occurs. A realm that is included in a realm's list of static accounting servers and specified in `proxy.ini` as doing static accounting, it gets duplicate accounting packets.

Proxy RADIUS Configuration (.pro) File

For each Proxy RADIUS realm that you want to configure on the Steel-Belted Radius server, you must create a file called `RealmName.pro`, where `RealmName` is the name of the realm, and you must add this `RealmName` to the `[Realms]` section of the `proxy.ini` file.

If you create or edit a `RealmName.pro` file, you can apply your configuration changes dynamically, without stopping the server. Depending on your operating system:

- Under **UNIX**: Simply issue the HUP signal to the Steel-Belted Radius process:
kill -HUP ProcessID

- Under **Windows**: Run the RADHUP.EXE program from the command shell. (RADHUP.EXE is located in the server directory that you specified at installation time, usually C:\RADIUS\Service.)

After you do this, Steel-Belted Radius re-reads proxy.ini, filter.ini, and all .pro and .dir files in the server directory, and resets its realm configuration accordingly.

*Note: Rarely, you must edit radius.ini while configuring a realm. If you do edit radius.ini, you **must** stop and restart the Steel-Belted Radius before your new configuration is fully loaded.*

Proxy RADIUS [Auth] Section

The [Auth] section of a *RealmName*.pro file configures authentication for the Proxy RADIUS realm. The key parameters in these sections are:

- `TargetsSection`, which names the target selection strategy you want to use.
- `FilterIn` and `FilterOut`, which name the attribute filters you want applied to request and response packets, respectively.

The following table lists the fields that may be present in a .pro file's [Auth] section.

Realm Configuration	
[Auth] Field	Meaning
<code>Enable=<i>n</i></code>	If set to 1, the Enable field enables forwarding of authentication packets to the realm called <i>RealmName</i> . If the Enable field is set to 0, the realm called <i>RealmName</i> is disabled for authentication.
<code>FilterIn=<i>name</i></code>	<i>name</i> is the name of a filter, a section in the filter.ini file called [<i>name</i>]. This section specifies rules for editing the attributes in a response packet as it returns "in" from the Proxy RADIUS realm, before relaying the packet back to the RADIUS client. See "FilterIn" on page 281.
<code>FilterOut=<i>name</i></code>	<i>name</i> is the name of a filter, a section in the filter.ini file called [<i>name</i>]. This section specifies rules for editing the attributes in a RADIUS request packet before forwarding the packet "out" to a Proxy RADIUS realm. See "FilterOut" on page 281.
<code>MessageAuthenticator=<i>n</i></code>	If set to 1, a Message-Authenticator is inserted into each packet forwarded to any target server in the realm. The default value is 0. <i>NOTE: both the proxy and the target RADIUS server requires this functionality.</i>

Realm Configuration

[Auth] Field

Meaning

NumAttempts=*n*

n is the number of times a timeout may occur when attempting to contact servers within the realm, before a failure is declared and the attempts are stopped.

RequestTimeout=*x, y, z*

A list of times, in seconds, to wait when attempting to contact a target server before timing out. The first value is the time to wait before the first timeout, and so on.

The number of items in the list should be no greater than the NumAttempts setting. If NumAttempts is greater, than the last number listed is reused for subsequent timeouts.

NOTE: You can specify RequestTimeout or RequestTimeoutMills, but not both.

RequestTimeoutMills=*x, y, z*

A list of times, in milliseconds, to wait when attempting to contact a target server before timing out. The first value is the time to wait before the first timeout, and so on.

The number of items in the list should be no greater than the NumAttempts setting. If NumAttempts is greater, than the last number listed is reused for subsequent timeouts.

NOTE: You can specify RequestTimeout or RequestTimeoutMills, but not both.

RoundRobin=*n*

n is the number of target servers that are participating in "round-robin" load balancing. The count begins from the top of the list in the [name] section identified by TargetsSection. Other listed targets are used only after the round-robin targets fail for a particular request.

StripRealm=*n*

If set to 1, strip the realm name from the username before forwarding. If set to 0, name stripping is disabled.

NOTE: For Proxy RADIUS realms, realm name stripping is disabled (StripRealm is set to 0) by default. If you want to enable it, you must explicitly set StripRealm to 1.

TargetsSection=*name*

name identifies a section called [name] that appears elsewhere in the .pro file. This section lists all the targets in a Proxy RADIUS realm. When it receives a request for this Proxy RADIUS realm, Steel-Belted Radius selects a target from this list.

Having the TargetsSection field available in the [Auth] and [Acct] sections permits you to name different target selection parameters for Proxy RADIUS authentication and accounting.

The default value of name is Targets; in which case the name of the section is [Targets].

FilterOut

The `FilterOut=name` parameter causes Steel-Belted Radius to apply the filtering rules found in the `[name]` section of `filter.ini`. These rules are applied while Steel-Belted Radius is processing the incoming RADIUS request packet, and before it directs the packet “out” to the destination realm. You may also think of this as filtering various attributes and values “out” of the request before directing it to the realm.

Note: FilterOut only affects the processing of a request packet if attributes within the request (User-Name, Called-Station-Id) indicate that the request should be routed to a realm.

FilterIn

The `FilterIn=name` parameter causes Steel-Belted Radius to apply the filtering rules found in the `[name]` section of `filter.ini`. These rules are applied after Steel-Belted Radius has received a response “in” from the destination realm, and while it is preparing the RADIUS response packet for its client. You may also think of this as filtering various attributes and values “in” to the response before returning it to the client.

Note: FilterIn only affects the response packet if a realm was used to process the request.

Proxy RADIUS [Acct] Section

The `[Acct]` section configures accounting. The key parameters in these sections are:

- `TargetsSection`, which names the target selection strategy you want to use.
- `FilterIn` and `FilterOut`, which name the attribute filters you want applied to request and response packets, respectively.

The following table lists the fields that may be present in a `.pro` file’s `[Acct]` section.

Realm Configuration

[Acct] Field	Meaning
Block= <i>n</i>	<p>If set to 0, the NAS sends an accounting acknowledgement immediately (for example, after Steel-Belted Radius records an accounting message). If the Block field is set to 1 (the default), the NAS waits for a response from the target realm before sending an accounting acknowledgement.</p> <p><i>NOTE: Set the Block field to 0 if your NAS times out waiting for a response from the target realm.</i></p>
Enable= <i>n</i>	<p>If set to 1, the Enable field enables forwarding of accounting packets to the realm called <i>RealmName</i>. If the Enable field is set to 0, the realm called <i>RealmName</i> is disabled for accounting.</p>
FilterIn= <i>name</i>	<p><i>name</i> is the name of a filter, a section in the filter.ini file called [<i>name</i>]. This section specifies rules for editing the attributes in a response packet as it returns “in” from the Proxy RADIUS realm, before relaying the packet back to the RADIUS client.</p> <p>See “FilterIn” on page 281.</p>
FilterOut= <i>name</i>	<p><i>name</i> is the name of a filter, a section in the filter.ini file called [<i>name</i>]. This section specifies rules for editing the attributes in a RADIUS request packet before forwarding the packet “out” to a Proxy RADIUS realm.</p> <p>See “FilterOut” on page 281.</p>
NumAttempts= <i>n</i>	<p><i>n</i> is the number of times a timeout may occur when attempting to contact servers within the realm, before a failure is declared and the attempts are stopped.</p>
RecordLocally= <i>n</i>	<p>If set to 1, log the packet locally before forwarding. If set to 0, forward the packet and do not log locally.</p>
RequestTimeout= <i>x, y, z</i>	<p>A list of times, in seconds, to wait when attempting to contact a target server before timing out. The first value is the time to wait before the first timeout, and so on.</p> <p>The number of items in the list should be no greater than the NumAttempts setting. If NumAttempts is greater, than the last number listed is reused for subsequent timeouts.</p> <p><i>NOTE: You can specify RequestTimeout or RequestTimeoutMills, but not both.</i></p>

Realm Configuration

[Acct] Field	Meaning
RequestTimeoutMills= <i>x, y, z</i>	<p>A list of times, in milliseconds, to wait when attempting to contact a target server before timing out. The first value is the time to wait before the first timeout, and so on.</p> <p>The number of items in the list should be no greater than the NumAttempts setting. If NumAttempts is greater, than the last number listed is reused for subsequent timeouts.</p> <p><i>NOTE: You can specify RequestTimeout or RequestTimeoutMills, but not both.</i></p>
RoundRobin= <i>n</i>	<p><i>n</i> is the number of target servers that are participating in “round-robin” load balancing. The count begins from the top of the list in the [name] section identified by TargetsSection. Other listed targets are only used after the round-robin targets fail for a particular request.</p>
StaticAcctRealms	<p>If a setting is supplied for this field, accounting packets are forwarded to a list of realms. The setting given must be a section name defined in the proxyrl.ini file that lists the realms to which the accounting packets should be forwarded.</p> <p>See “Proxyrl.ini” on page 277.</p>
StripRealm= <i>n</i>	<p>If set to 1, strip the realm name from the username before forwarding. If set to 0, name stripping is disabled.</p> <p><i>NOTE: For Proxy RADIUS realms, realm name stripping is disabled (StripRealm is set to 0) by default. If you want to enable it, you must explicitly set StripRealm to 1.</i></p>
TargetsSection= <i>name</i>	<p><i>name</i> identifies a section called [name] that appears elsewhere in the .pro file. This section lists all the targets in a Proxy RADIUS realm. When it receives a request for this Proxy RADIUS realm, Steel-Belted Radius selects a target from this list.</p> <p>Having the TargetsSection field available in the [Auth] and [Acct] sections permits you to name different target selection parameters for Proxy RADIUS authentication and accounting.</p> <p>The default value of name is Targets; in which case the name of the section is [Targets].</p>

FilterOut

The FilterOut=*name* parameter causes Steel-Belted Radius to apply the filtering rules found in the [name] section of filter.ini. These rules are applied while Steel-Belted Radius is processing the incoming RADIUS request packet, and before it directs the packet “out” to the destination realm. You may also think of this as

filtering various attributes and values “out” of the request before directing it to the realm.

Note: FilterOut affects only the processing of a request packet if attributes within the request (User-Name, Called-Station-Id) indicate that the request should be routed to a realm.

FilterIn

The `FilterIn=name` parameter causes Steel-Belted Radius to apply the filtering rules found in the `[name]` section of `filter.ini`. These rules are applied after Steel-Belted Radius has received a response “in” from the destination realm, and while it is preparing the RADIUS response packet for its client. You may also think of this as filtering various attributes and values “in” to the response before returning it to the client.

Note: FilterIn affects only the response packet if a realm was used to process the request.

Proxy RADIUS [AutoStop] Section

The `[AutoStop]` section of a realm configuration file permits you to activate the Proxy AutoStop feature. When this feature is enabled, an AutoStop request is automatically recorded and associated with the session in the current sessions database when the initial Accounting-Start message is received. This AutoStop message may be used later to simulate an Accounting-Stop message which is fed back into the request processing engine, causing it to be forwarded to the appropriate realms and for the normal processes of ending the user session to be enacted.

Note: As the AutoStop record is generated when the session begins, it is simply a duplicate of the original Start request and does not have access to information about the lifetime of the user’s actual activity.

Warning: AutoStop records are not saved on persistent storage: this means that if Steel-Belted Radius is restarted, this information is lost and hence Accounting-Stop messages cannot be simulated for these user sessions.

See “Proxy AutoStop Feature” on page 71.

The following field can be configured in the [AutoStop] section:

Realm Configuration	
[AutoStop] Field	Meaning
Enable	Set to 0 to disable AutoStop, or 1 to enable AutoStop for this particular realm. The default value is 0.

In addition to this flag, you must also enable (set to 1) the following fields for AutoStop to operate:

File	Section	Field
<i>RealmName.pro</i>	[Acct]	Enable RecordLocally
radius.ini	[Configuration]	AcctAutoStopEnable

Proxy RADIUS [Called-Station-ID] Section

The [Called-Station-ID] section of a *RealmName.pro* file allows the target realm to be selected based on DNIS. The [Called-Station-ID] section lists each DNIS string that identifies the realm. If this string is found in the Called-Station-Id attribute of an incoming RADIUS request, then the request is assumed to be addressed to this realm.

The syntax is as follows:

```
[Called-Station-ID]
String
String
.
.
.
```

where *String* is a DNIS string.

For example:

```
[Called-Station-ID]
8005551212
8005551213
6175551212
```

You can also use wildcards, as in the following example:

```
[Called-Station-ID]
800*
```

Proxy RADIUS Target Selection Rules

Each `[name]` section of a `RealmName.pro` file specifies a set of rules that Steel-Belted Radius can use to select a target for proxy-forwarding within the Proxy RADIUS realm. Each `[name]` section consists of a list of target servers. For any particular request, if the first listed server fails to respond (or is presumed down), then the other servers are tried in the order listed. A `[name]` section is activated by referencing it from the `[Auth]` and/or `[Acct]` sections.

To activate...	Use...
a <code>[name]</code> section for authentication	<code>TargetName=name</code> in the <code>[Auth]</code> section
the same <code>[name]</code> section for accounting	<code>TargetName=name</code> in the <code>[Acct]</code> section
some <code>[other]</code> section for accounting	<code>TargetName=other</code> in the <code>[Acct]</code> section

The full syntax is as follows:

```
[Auth]
TargetsSection=nameB
```

```
[Acct]
TargetsSection=nameA
```

```
[nameA]
Server = n
Server = n
.
.
.
```

```
[nameB]
Server = n
Server = n
.
.
.
```

where `Server` is the name of a server that you've configured as a target for standard Proxy RADIUS forwarding, and `n` is explained in the next section.

`Server` must match a Proxy entry in the Steel-Belted Radius database. This Proxy entry provides the address and shared secret for the target server. All other settings in the Proxy entry (retry policy, proxy accounting) are overridden by the settings that you configure in the `RealmName.pro` file.

Note: If your server has multiple interface cards, you may add a parameter referring to the interface to each line to order the outgoing proxy traffic for the realm through a particular interface. See “proxy.ini [Interfaces] Section” on page 274.

Round-Robin Load Balancing

If you have multiple target servers in a realm, you can select whether to use them in round-robin fashion (load balancing), primary/backup fashion, or a combination of both. The value of the RoundRobin entry in the [Auth] or [Acct] section indicates the number of targets that are to be used in round-robin fashion. The count begins from the top of list in the [name] section. Other listed targets are used only if the round-robin targets fail for a particular request. If RoundRobin is 0 or 1, all requests are routed to the first target in the [name] list, assuming that it is up, then the others are tried in the order listed.

If RoundRobin is 2 or greater (say, n), then each request is routed to a different target server, in rotation among the first n listed targets. Requests are thus load-balanced evenly among those targets. For any particular request, if one target fails to respond, other targets are attempted. The round-robin targets are tried first; if they all fail to respond, any additional targets are then tried in the order in which they appear in the list.

In the following example, RoundRobin is 3. Under normal circumstances, requests are balanced in round-robin fashion among the first three targets. The first request goes to Bert; the next goes to Ernie; the next to George; the next to Bert; the next to Ernie; the next to George; and so on. If any of these servers go down at some point, the other two are tried, in list order. The fourth target (Mary) receives requests only when other targets are down.

```
[Auth]
RoundRobin=3
NumAttempts=8
TargetsSection=Targets
```

```
[Targets]
Bert=1
Ernie=1
George=1
Mary=5
```

Selecting a Backup Server

If `RoundRobin` is set to 0, Steel-Belted Radius makes a selection from the “other” servers in the list only if the primary server is down.

For example:

```
[Auth]
RoundRobin=0
NumAttempts=8
TargetsSection=Targets

[Targets]
Bert=1
Ernie=1
```

In this case, Bert is used until there is a problem; then Ernie becomes the server of second choice.

Realm Retry Policy

Each target selection rule in the `[name]` section permits you to name a target and assign it a numeric value:

```
[name]
Server = n
Server = n
.
.
.
```

The `n` field indicates the number of times to retry requests to this target server when it doesn't respond (when no response is received from the server within the amount of time set by `RequestTimeout` in the `[Auth]` or `[Acct]` section).

The total number of attempts to all servers within the entire realm is given by the `NumAttempts` value in the `[Auth]` or `[Acct]` section. For example, let's say that `NumAttempts` is 8 and there are three target servers, each with `n` set to 3:

```
[Auth]
NumAttempts=8
TargetsSection=Targets

[Targets]
Bert=3
Ernie=3
George=3
```

Let's say that all three servers are down when a request comes into the realm. The first target (Bert) is tried 3 times; then the second target (Ernie) is tried 3 times;

and the third target (`George`) is tried 2 times. At this point, the total number of tries to all servers in the realm is 8, which equals `NumAttempts`. Steel-Belted Radius returns a failure response from the realm.

Note: A third attempt to George could not be made unless you edited the `RealmName.pro` file, increased `NumAttempts` to 9, and reloaded Steel-Belted Radius.

Proxy RADIUS [FastFail] Section

The [FastFail] section of a realm configuration file permits you to fine-tune retry policies for individual realms, and for specific targets within a realm. If you provide a [FastFail] section, the `ProxyFastFail` parameter in the `radius.ini` [Configuration] section is ignored.

The following fields may be present in a [FastFail] section.

.pro File	
[FastFail] Field	Meaning
<code>MinFailures=x</code> <code>MinSeconds=y</code>	<p>These parameters define a tolerance level for failures to reach a target server within a realm. Such “failures” are judged according to the <code>NumAttempts</code> and <code>RequestTimeout</code> settings that you defined in the [Auth] or [Acct] sections.</p> <p>A target is presumed down once <i>x</i> consecutive failures have occurred and at least <i>y</i> seconds have elapsed.</p> <p>Once a target is presumed down, Steel-Belted Radius directs proxy requests to another target in the same realm, if available. It does not wait for responses from the failed target.</p> <p>However, it sends strobe requests periodically to the failed target to detect when that server comes back up. Once a response is received to one of these strobe requests, that server is no longer presumed down.</p> <p><i>NOTE: Strobe requests are sent to the “down” target server only if there are proxy requests addressed to its realm.</i></p>
<code>ResetSeconds=z</code>	<p>Once the realm's tolerance level is exceeded, this parameter specifies how long a target may be presumed down.</p> <p>The <code>ResetSeconds</code> value indicates the maximum number of seconds during which a server can be presumed down in the absence of strobe requests. If <i>z</i> seconds elapse with no strobe requests sent to the down server, the server is reset to “up.”</p> <p>Thus, the status of a target that is presumed down is reset to “up” when one of the following occurs: (1) A response to a strobe request is received from the server (2) There has been no request sent to the server for <i>z</i> seconds.</p>

Proxy RADIUS [ModifyUser] Section

The [ModifyUser] section of a realm configuration file permits you to decorate a realm, where the realm is determined by other means, such as DNIS or attribute mapping.

This is used mainly to enhance directed realms. For example, the following two users are in the database: `george@gm` and `george@ford`. Either user could log in as `george`, as Steel-Belted Radius would determine the realm, for example, by DNIS. Based on the realm, Steel-Belted Radius would append either `@gm` or `@ford` to the user name, and then use the Native User directed method to authenticate.

This methodology could also be used in a double-proxy situation. The first proxy uses DNIS to determine a realm, then decorates the name and forwards it to the next hop server. This second proxy (which may be a legacy RADIUS server that doesn't understand DNIS) could then handle realms based on the name decoration.

The following fields may be present in a [ModifyUser] section:

.pro File	
[ModifyUser] Field	Meaning
<code>AddPrefix=prefix</code>	These parameters define the User-Name prefix and suffix.
<code>AddSuffix=suffix</code>	

Proxy RADIUS [SpooledAccounting] Section

Proxy Spooling is configured within the [SpooledAccounting] section of a *RealmName.pro* file. It has the following fields:

RealmName.pro	
[SpooledAccounting]	
Field	Meaning
Enable	Set to 1 to enable Proxy Spooling. The default value is 0.
RolloverSeconds	The rollover interval in seconds. After the interval elapses, the current spool file is closed and a new one is created. The default value is 600 (10 minutes.)
RolloverSize	The rollover file size limit in bytes. After the file size exceeds this limit, the current spool file is closed and a new one is created. If both RolloverSeconds and RolloverSize are set, the first field that exceeds its limit initiates rollover. The default value is 1,048,576 bytes (1 megabyte).

RealmName.pro
[SpooledAccounting]

Field	Meaning
Directory	<p>The directory where the spool (.psf) files is stored. The directory must be manually created in the RADIUS service directory.</p> <p>The default value is <code>.\RealmName</code></p> <p>Important: Each realm must have its own directory for spool files. Otherwise, packets for multiple realms would be interspersed and a problem in one realm could prevent subsequent packets to other realms from being forwarded.</p>
RetryInterval	<p>The interval in seconds prior to retrying a proxy request if the target system (the downstream server where accounting data for this realm is sent) is down.</p> <p>The default value is 60.</p>
ShutdownDelay	<p>The amount of time (given as the number of seconds) prior to the execution of a shutdown request during which the final undelivered spooled packets in the spool file can be sent to their target. This value should be set according to the amount of accounting data normally received for this realm, and other relevant network conditions.</p> <p>If the target system is down when Steel-Belted Radius shuts down, this setting is not applied, and unspooling terminates immediately (and Steel-Belted Radius shuts down immediately). Upon restart, unspooling of accounting data restarts from the beginning of the oldest spool file.</p> <p>The default value is 20.</p>

For example:

```
[SpooledAccounting]
Enable=1
RolloverSeconds=600
RolloverSize=1048576
Directory=.\all_acct_data
RetryInterval=60
ShutdownDelay=20
```

Important: Do not enable proxy spooling for realms that not enabled for accounting.

Retry Sequence

If Steel-Belted Radius receives an accounting packet for a realm, and the target system is down, Steel-Belted Radius implements the *RealmName.pro* retry configuration, as in the following example:

```
[Acct]
```

```
RequestTimeout=5, 3, 5
NumAttempts=3
```

In this example, Steel-Belted Radius attempts to proxy forward the accounting packet to the target IP address, as it would in a non-SpooledAccounting scenario. Three attempts are made; the first waits for five seconds before timing out, the second three seconds, and the third five seconds.

If there is still no response from the target after three attempts, the `RetryInterval` in the `[SpooledAccounting]` section is applied. If `RetryInterval` equals 60, then five seconds after the last unsuccessful `NumAttempts` is completed, Steel-Belted Radius waits another sixty seconds and then attempts the entire retry policy again.

Directed Realm Configuration (.dir) File

A *directed realm* specifies target methods for directed authentication and/or directed accounting. Its realm configuration file is called *RealmName.dir*.

The *directed authentication* feature permits the server to bypass its Authentication Methods list and map an incoming RADIUS request to one or more specific authentication methods. Steel-Belted Radius chooses the destination method based on routing information found in the request packet. The destination methods may be any authentication methods already configured on the local Steel-Belted Radius server, regardless of how they were configured; for example, a method may have been configured using the Administrator dialogs, the LDAP configuration interface, or an `.aut` configuration file.

If no directed authentication method is configured, every request percolates through the same Authentication Methods list, as defined in the Administrator program's Configuration dialog. This behavior may or may not be ideal for every customer. Directed authentication allows you to tailor an authentication methods list to a customer's exact needs.

Directed accounting is also possible. The destination accounting method may be the Steel-Belted Radius accounting log, an external database configured using an `.acc` file, or a distinct accounting log file that contains entries only for this customer.

To activate these features, you must create *RealmName.dir* files, place them in the Steel-Belted Radius server directory, and list them in the `[Directed]` section of `proxy.ini`. Subsequently, any requests that arrive addressed to one of these realm names are processed on the local server using the instructions you've provided in `proxy.ini` and in the corresponding *RealmName.dir* file.

After you edit a *RealmName.dir* file, you must apply your changes as follows. If you have added or changed:

- Any directed accounting methods at all, you must stop and restart the server to load your new configuration.
- Directed authentication methods in which external database (SQL or LDAP) authentication is used, you must stop and restart the server to load your new configuration.
- Directed authentication methods in which local or pass-through (Native, UNIX, Domain, Host, SecurID, or TACACS+) authentication is used, you can apply your configuration changes dynamically, without stopping the server. Depending on your operating system:
 - Under **UNIX**: Simply issue the HUP signal to the Steel-Belted Radius process:
kill -HUP ProcessID
 - Under **Windows**: Run the RADHUP.EXE program from the command shell. (RADHUP.EXE is located in the server directory that you specified at installation time, usually C:\RADIUS\Service.)

Steel-Belted Radius re-reads proxy.ini, filter.ini, and all .pro and .dir files in the server directory, and resets its realm configuration accordingly.

Note: Rarely, you must edit radius.ini while configuring a realm. If you edit radius.ini, you must stop and restart Steel-Belted Radius before your new configuration is fully loaded.

Directed Realm [Auth] Section

Directed authentication is enabled in a realm by setting the Enable parameter in the [Auth] section of the corresponding *RealmName.dir* file, where *RealmName* is the name of the realm. The syntax is as follows:

```
[Auth]
Enable = n
StripRealm = n
```

where the fields have meaning as follows:

Directed Realm [Auth] Field	Meaning
Enable= <i>n</i>	<p>If Enable is set to 1 in the [Auth] section of a <i>RealmName.dir</i> file, the directed authentication realm called <i>RealmName</i> is enabled. If set to 0, the realm is disabled.</p> <p>By enabling a directed authentication realm, you make it possible for Steel-Belted Radius to override the Authentication Methods list on the local server by providing an alternate list - for requests addressed to this realm only. Details of this list are provided in the [AuthMethods] section of the same <i>RealmName.dir</i> file.</p>
StripRealm= <i>n</i>	<p>If StripRealm is set to 1, Steel-Belted Radius strips the realm name from the username before attempting to authenticate the user's request.</p> <p>If set to 0, realm name stripping is disabled.</p> <p><i>NOTE: For directed realms, realm name is enabled (StripRealm is set to 1) by default. If you want to disable it, you must explicitly set StripRealm to 0.</i></p>

Directed Realm [AuthMethods] Section

If directed authentication is enabled, the [AuthMethods] section of a *RealmName.dir* file lists one or more authentication methods to be used.

The syntax is as follows:

```
[AuthMethods]
  Description
  Description
  .
  .
  .
```

where *Description* is the “official name” of an authentication method already configured on the Steel-Belted Radius server. For example:

- The names of internal authentication methods should be well known to you from the Configuration dialog’s Authentication Methods list. Any of these names may be used as a Description string in a *RealmName.dir* [AuthMethods] section. For example:

```
Native User
Domain User   (Windows only)
Domain Group  (Windows only)
Host User     (Windows only)
Host Group    (Windows only)
```

```
SecurID User
SecurID Prefix
SecurID Suffix
UNIX User      (UNIX only)
UNIX Group     (UNIX only)
TACACS+ User
TACACS+ Prefix
TACACS+ Suffix
```

- Similarly, if you want your [AuthMethods] section to reference external authentication methods, your *Description* strings must match the names of these methods. For a Proxy RADIUS target server, this is the string entered in the **Forward to** field in the Proxy dialog. For an external database, this is the InitializationString value from the [Bootstrap] section of the corresponding .aut file.

Note: There is no interaction between the settings in the Configuration dialog and in RealmName.dir files, or between different RealmName.dir files. For example, if you disable the User method (for UNIX) or Domain User method (for Windows) in the Configuration dialog while it is enabled in a RealmName.dir file, it remains enabled in RealmName.dir.

Directed Realm [Acct] Section

Directed accounting is enabled in a realm by setting the Enable parameter in the [Acct] section of the corresponding *RealmName.dir* file, where *RealmName* is the name of the realm. The syntax is as follows:

```
[Acct]
Enable = n
StripRealm = n
RecordLocally = n
```

where the fields have meaning as follows:

Directed Realm [Acct] Field	Meaning
Enable= <i>n</i>	<p>If Enable is set to 1 in the [Acct] section of a <i>RealmName.dir</i> file, the directed accounting realm called <i>RealmName</i> is enabled. If set to 0, the realm is disabled.</p> <p>By enabling a directed accounting realm, you make it possible for Steel-Belted Radius to override the normally configured accounting methods on the local server by providing an alternate list - for requests addressed to this realm only. Details of this list are provided in the [AcctMethods] section of the same <i>RealmName.dir</i> file.</p>
RecordLocally= <i>n</i>	<p>If RecordLocally is set to 1, Steel-Belted Radius writes accounting records to its main accounting log file in addition to the accounting destinations specified in [AcctMethods]. If RecordLocally is set to 0, this feature is disabled.</p>
StaticAcctRealms	<p>If a setting is supplied for this field, accounting packets are forwarded to a list of realms. The setting given must be a section name defined in the proxyrl.ini file that lists the realms to which the accounting packets should be forwarded.</p> <p>See "Proxyrl.ini" on page 277.</p>
StripRealm= <i>n</i>	<p>If StripRealm is set to 1, Steel-Belted Radius strips the realm name from the username before attempting to authenticate the user's request. If set to 0, realm name stripping is disabled.</p> <p><i>NOTE: For directed realms, username stripping is enabled (StripRealm is set to 1) by default. If you want to disable it, you must explicitly set StripRealm to 0.</i></p>

Directed Realm [AcctMethods] Section

If directed accounting is enabled, the [AcctMethods] section of a *RealmName.dir* file lists one or more accounting methods to be used.

The syntax is as follows:

```
[AcctMethods]
  Description
  Description
  .
  .
  .
```

where *Description* is the "official name" of a directed accounting method configured in the proxy.ini file.

See "proxy.ini [DirectedAcctMethods] Section" on page 273.

Directed Realm [Called-Station-ID] Section

The [Called-Station-ID] section of a *RealmName.dir* file allows Steel-Belted Radius to select a realm to be used for directed authentication and/or accounting based on DNIS information supplied in an incoming RADIUS packet. The [Called-Station-ID] section lists each DNIS string that identifies the realm. If this string is found in the Called-Station-Id attribute of an incoming request, the directed authentication and/or accounting rules found in the corresponding *RealmName.dir* file are applied to the request.

The syntax is as follows:

```
[Called-Station-ID]
String
String
.
.
.
```

where *String* is a DNIS string.

Directed Realm [ModifyUser] Section

The [ModifyUser] section of a realm directed file permits you to decorate a realm, where the realm is determined by other means, such as DNIS or attribute mapping.

This is used mainly to enhance directed realms. For example, the following two users are in the database: *george@gm* and *george@ford*. Either user could log in as *george*, as Steel-Belted Radius would determine the realm, for example, by DNIS. Based on the realm, Steel-Belted Radius would append either *@gm* or *@ford* to the user name, and then use the Native User directed method to authenticate.

The following fields may be present in a [ModifyUser] section:

.dir File	
[ModifyUser] Field	Meaning
<i>AddPrefix=prefix</i>	These parameters define the User-Name prefix and suffix.
<i>AddSuffix=suffix</i>	

Extensible Authentication Protocol

8

- EAP Concepts
- EAP Types
- eap.ini File
- Configuring For EAP-TTLS and EAP-PEAP
- Examples of EAP Usage

EAP Concepts

Steel-Belted Radius supports Extensible Authentication Protocol (EAP), a standard for communication between clients, Access Points, NASs, and servers that provides for the future extensibility of authentication protocols.

EAP allows specialized knowledge about authentication protocols to be taken out of a NAS or Access Point so that it is merely a conduit between authentication server and client. This means that new types of authentication can be supported by adding the appropriate functionality to server and client, without needing to make any changes to PPP or NAS devices. When the authentication process is complete, the RADIUS server simply informs the NAS or Access Point of the result.

For technical details about EAP, see RFC 2284, “PPP Extensible Authentication Protocol (EAP),” and RFC 2869, “RADIUS Extensions.”

Steel-Belted Radius supports several EAP authentication mechanisms, such as TTLS, TLS, PEAP, LEAP, MD5-Challenge, and Generic Token. Support for EAP has been designed to anticipate other innovative authentication types appearing on the horizon.

Note: EAP-TTLS and EAP-TLS are available as separate plug-in modules but require a specific license. They are described in more detail in a separate manual accompanying the license.

Handling EAP Requests

The flow of RADIUS packets in an EAP scenario is quite different from the transactions using standard user credentials (e.g. PAP, CHAP). Standard user credentials involve the transmission of a RADIUS request from the NAS (or Access Point) to Steel-Belted Radius and a response (either an Accept or Reject) from the server back to the NAS (or Access Point).

With EAP, the first packet sent from the NAS (or Access Point) to Steel-Belted Radius contains an `EAP-Message` attribute containing an EAP Identity Response. This is a signal sent by the system being authenticated that it wants to be authenticated via EAP. It is now up to Steel-Belted Radius to select the EAP protocol with which it is to authenticate the end-user.

The contents of the `User-Name` attribute is the only guideline available to Steel-Belted Radius in selecting the appropriate EAP protocol. Should Steel-Belted Radius select an EAP protocol that is not supported by the client, the client has the opportunity to send an EAP-NAK and to request a specific alternate protocol.

Note: Given this general flow, a RADIUS request with EAP credentials must incur a minimum of two network round-trips between the NAS (or Access Point) and the Steel-Belted Radius before reaching a successful conclusion.

Automatic EAP Helpers

Automatic EAP helpers serve as intermediaries between EAP and traditional authentication methods. These helper modules may be configured (using an associated .eap file) to work with existing authentication methods to shield the authentication methods from the particulars of the selected EAP protocol.

The following table reveals whether each EAP type is implemented as an EAP helper or stand-alone module in Steel-Belted Radius:

EAP-Type	Implemented As
LEAP	Automatic EAP helper for MS-CHAP-v1
EAP Generic-Token	Authentication Method Module (SecurID)
EAP MD5-Challenge	Automatic EAP helper for CHAP

Whether an automatic EAP helper can be used in conjunction with a specific authentication method depends on what types of credentials the authentication method supports.

The automatic EAP helper that implements EAP MD5-Challenge generates CHAP credentials, while the helper that implements LEAP generates MS-CHAP-v1 credentials. As such, EAP MD5-Challenge can be used only with authentication methods that support CHAP, and LEAP can be used only with authentication methods that support MS-CHAP-v1.

The following table summarizes the support for MS-CHAP-v1 and CHAP in the Steel-Belted Radius authentication methods.

Authentication Method	MS-CHAP-V1	CHAP
LDAP	Yes for BindName (password must be stored in the clear in LDAP server), No for Bind	Yes for BindName (password must be stored in the clear in LDAP server), No for Bind
Native	Yes	Yes
Proxy RADIUS	Yes	Yes
SecurID	No	No
SQL	Yes if password is in clear in SQL database	Yes if password is in clear in SQL database
TACACS+	No	Yes
UNIX User	No	No

Authentication Method	MS-CHAP-V1	CHAP
UNIX Group	No	No
NT Domain User	Yes (server must be running under SYSTEM account on a pdc or bdc)	No
NT Domain Group	Yes (server must be running under SYSTEM account on a pdc or bdc)	No
Windows Domain User	Yes (server must be running under SYSTEM account)	No
Windows Domain Group	Yes (server must be running under SYSTEM account)	No

Authentication Request Routing

The order in which authentication methods and automatic EAP helpers are called to handle an authentication request depends on two factors:

- 1 The ordered list of enabled authentication methods (viewable in the Configuration panel of the Steel-Belted Radius administration GUI)
- 2 The EAP-related configuration for each of the enabled authentication methods (found in the `eap.ini` file).

When Steel-Belted Radius receives an authentication request that does not contain EAP credentials, it passes the request to each enabled authentication method until one of the methods claims the request.

The EAP settings in the `eap.ini` file come into play only when a request with EAP credentials is received. An authentication request contains EAP credentials if it includes one or more `EAP-Message` attributes and contains no other form of user credentials (e.g., `User-Password`).

EAP-Only Setting

When an authentication method's `EAP-Only` setting is 1, Steel-Belted Radius prevents the authentication method from being called for any request that does not contain EAP credentials. Under this setting, the authentication method is also bypassed if an authentication request specifically requests an EAP protocol that is not listed in the authentication method's `EAP-Type` list in the `eap.ini` file.

First-Handle-Via-Auto-EAP Setting

If your configuration involves clients using more than one EAP protocol, the Steel-Belted Radius server must select an initial EAP protocol with which to proceed when receiving an authentication request with EAP credentials.

Selecting the incorrect EAP protocol is not fatal; the client simply sends an EAP NAK in response to the server's selected protocol and suggests an alternate one. After one additional network round-trip, the correct EAP protocol becomes active.

Depending on the capabilities of the authentication methods being used, you may be able to cut out this additional network round-trip that affects a portion of your EAP-based authentication requests.

If an authentication method can check for the existence of a user and can retrieve the user's password information with only the information available in the authentication request (e.g., the username), it is said to be *prefetch-capable*. A prefetch-capable authentication method could be consulted first to see if a user exists in its database before committing to a specific EAP protocol.

If your authentication method is prefetch-capable, you would set `First-Handle-Via-Auto-EAP` to 0, indicating that the authentication method should have the first chance to handle the request. You would also set `First-Handle-Via-Auto-EAP` to 0 if the authentication method is capable of handling EAP credentials all on its own (clearly, it would not expect an automatic helper EAP method to do work on its behalf in this case).

By configuring the authentication method to be called first, Steel-Belted Radius can delay selection of an EAP protocol until it has ascertained whether the user exists in a particular authentication method's database. This is a useful technique when you plan to use more than one EAP protocol, but you don't know which one the client will want. Even in this scenario, automatic EAP helpers may still end up performing the EAP protocol processing; they will take over after the authentication method has retrieved a user's password information, rather than before.

The goal of an automatic EAP helper is to generate credentials against which traditional authentication methods (ones that do not understand EAP) can operate. Once an automatic EAP helper has generated these credentials, the authentication method that triggered the use of the helper is checked first for a password/credential match. Should this match not be present, the same traditional credentials are passed to all remaining enabled authentication method in the master list (in the order in which they appear in the list).

Auth. Method	Prefetch Capable?
LDAP	Yes, if using BindName (rather than the Bind option)
Native User	Yes
NT Domain	No
SQL	Yes, if password does not need to be used as an input parameter in the SQL statement
UNIX User	No
Windows Domain	No

EAP-NAK Notifications

If you are supporting only one type of client or only one EAP protocol, Steel-Belted Radius selects that EAP protocol for all EAP-based authentication requests it receives. If you are planning to support multiple EAP protocols and don't intend to maintain databases that track the appropriate EAP protocol on a user-by-user basis, Steel-Belted Radius automatically selects the appropriate EAP protocol for you.

When multiple EAP protocols are in play, you should configure each authentication method you plan to use with all the EAP protocols that may be used with it. In this configuration, when Steel-Belted Radius receives an authentication request containing EAP information, it chooses the first EAP protocol listed for the first authentication method that claims the request. Should the client require a different EAP protocol, it sends back an EAP-NAK that specifies the EAP protocol it would prefer to use.

After receiving an EAP-NAK, Steel-Belted Radius performs a scan of the authentication methods, in search of the first authentication method that has the requested EAP protocol listed (the authentication method may support this EAP protocol directly or with the help of an automatic EAP helper).

If the requested EAP protocol does not appear in any of the authentication methods' lists of supported EAP protocols, Steel-Belted Radius rejects the authentication request.

Reauthenticating Connections

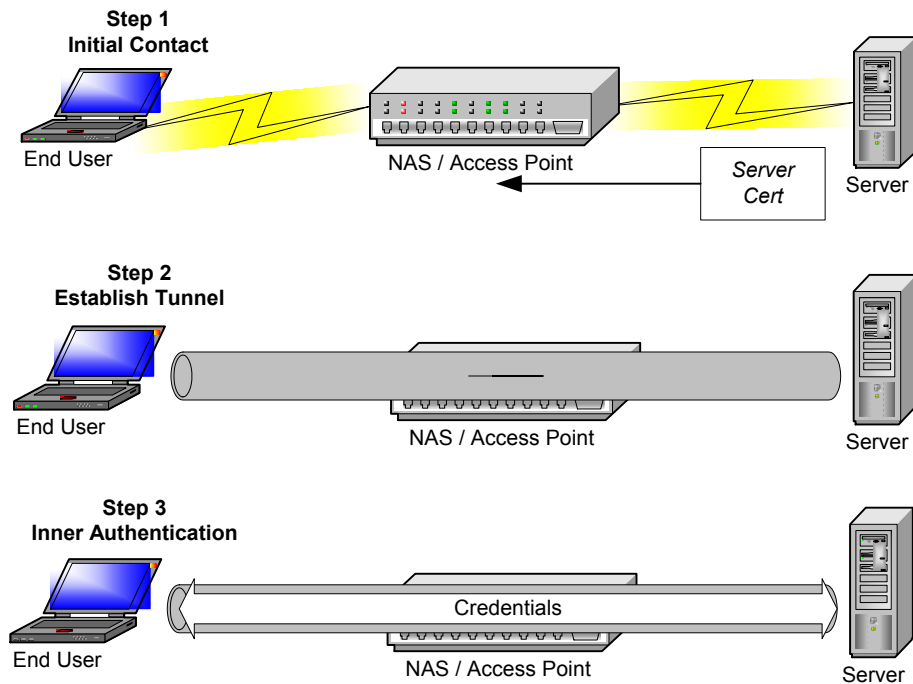
Most Access Points understand only a limited number of attributes that may be included in a RADIUS response to signal that the user has been accepted. The `Session-Timeout` attribute is of particular significance in a WLAN realm as it instructs the Access Point how long to allow the user to remain connected to a WLAN before having to re-authenticate to the Steel-Belted Radius server.

You can configure your choice of `Session-Timeout` settings using standard Steel-Belted Radius reply-list items on a user-by-user basis.

Note: Not all Access Points support the Session-Timeout attribute. You should check your Access Points' specifications to determine whether this configuration must be performed in a fixed manner on the Access Point or if the Access Point should defer to the server.

EAP Types

EAP-PEAP



LEAP

LEAP is an EAP protocol devised by Cisco that allows a client and server to mutually authenticate each other. Each party does this by confirming that the other has the MD4 hash of the user's password.

LEAP also supports the generation of keying material for use in link layer encryption via protocols such as WEP.

EAP Generic-Token

EAP Generic-Token is an EAP protocol that is defined as part of the base EAP specification. It allows for an open-ended exchange between the owner of a security token (e.g., SecurID) and an authentication server.

EAP MD5-Challenge

EAP MD5-Challenge is an EAP protocol that is defined as part of the base EAP specification. It has security characteristics similar to those provided by CHAP credentials, except that in the case of EAP MD5-Challenge, a RADIUS server rather than a NAS generates the challenge bytes.

eap.ini File

The eap.ini configuration file allows you to configure what EAP authentication types are attempted for authenticating users against the different Steel-Belted Radius authentication methods.

Each authentication method that you want EAP authentication to be performed against must be configured within this eap.ini file.

This file must contain one section for each authentication method that you use. The name of the section must be the same as the filename of the authentication method's configuration file, without the .aut extension. For example, if the name of .aut file is ldapauth.aut, the section in the eap.ini file should be identified as [ldapauth].

Each section contains the following fields:

<Authent-Method> field	Meaning
EAP-Only	If set to 0, the authentication method accepts all types of user credentials. This field should be set to 1 if the authentication method is given only EAP credentials or acts only as a back-end server to an automatic EAP protocol method. For authentication methods expected to handle EAP-TTLS inner authentications, this field should be set to 0 or 1 depending on the type of credentials used in the inner authentication. The default is 0.

<Authent-Method>	Meaning
field	
EAP-Type	<p>A comma-separated list of all the EAP protocols to support for this authentication method. The first protocol in the list is the primary protocol. Protocols that appear later in the list are used with this authentication method only if the client responds with an EAP NAK and specifies such a protocol or if another authentication method triggers the use of the protocol but cannot complete the request.</p> <p>Valid values include the following: LEAP, Generic-Token, and MD5-Challenge</p> <p>Leave this list empty to disable EAP for this authentication method.</p>
First-Handle-Via-Auto-EAP	<p>If set to 1 and the user credentials are EAP, an appropriate automatic EAP helper method is called before the authentication method. The purpose of calling the automatic EAP helper method is to convert the user's EAP credentials into a format acceptable to the authentication method.</p> <p>If set to 0, the authentication method itself handles the request directly, before any automatic helper methods.</p> <p>The default is 1.</p>

Example:

```
[Native-User]
EAP-Only = 0
First-Handle-Via-Auto-EAP = 0
EAP-Type = LEAP, MD5-Challenge

[NT-Domain]
EAP-Only = 0
EAP-Type = TTLS, LEAP
First-Handle-Via-Auto-EAP = 1

[NT-Host]
EAP-Only = 0
EAP-Type = LEAP
First-Handle-Via-Auto-EAP = 1
```

Note: Steel-Belted Radius comes configured with an eap.ini file that should work for all but the most complex or unusual environments. We suggest that you use this default configuration unless you find that it does not meet your needs.

Configuring For EAP-TTLS and EAP-PEAP

The EAP-TTLS protocol is supported in Steel-Belted Radius via a plug-in named `ttlsauth`, and EAP-PEAP via a plug-in named `peapauth`. EAP-TTLS and EAP-PEAP are authentication methods and, as such, appear in the administrative GUI much as a normal authentication method would. They are configured with nearly the same settings because they operate via a similar mechanism. The `ttlsauth` and `peapauth` plug-ins must be configured with a server certificate and the accompanying private key.

Note: You can download a time-limited evaluation certificate tool from <http://www.funk.com/RegFiles/sbr30conf.asp>.

The inner authentication forwarded by the plug-ins is processed by an attribute filter that can be used to embellish the list of forwarded attributes. Any Access-Accept response may be processed by another attribute filter that can decide which attributes are forwarded on the final RADIUS Access-Accept for the EAP-TTLS or EAP-PEAP exchange.

In order for EAP-TTLS to operate correctly, the `[ttlsauth]` section of the `eap.ini` file should set `EAP-Type` to `TTLS` and `First-Handle-Via-Auto-EAP` to `0`.

Likewise for EAP-PEAP, the `[peapauth]` section of the `eap.ini` file should set `EAP-Type` to `PEAP` and `First-Handle-Via-Auto-EAP` to `0`.

Note: An inner authentication is treated by Steel-Belted Radius as an authentication request that is separate and distinct from the authentication request that constructs the tunnel. If, for instance, the inner authentication in an EAP-TTLS exchange includes PAP credentials, the authentication is not processed by any authentication method that is marked with `EAP-Only=1` in the `eap.ini` file.

ttlsauth.aut and peapauth.aut files

The EAP-TTLS and EAP-PEAP plug-ins are configured via the `ttlsauth.aut` and `peapauth.aut` files respectively. These configuration files are re-read each time the Steel-Belted Radius server gets a HUP signal.

Note that you must configure the `[Certificate]` section of `radius.ini` to specify the path to the file that describes the server's certificate. For more information, refer to "radius.ini [Certificate] Section" on page 209.

Server_Settings Section

The [Server_Settings] section allows you to configure the basic operation of the plug-in. It consists of the following fields:

[Server_Settings] field	Meaning
TLS_Message_Fragment_Length	<p>Set to the maximum size TLS message length that may be generated during each iteration of the TLS exchange.</p> <p>Anecdotal evidence suggests that some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).</p> <p>The default value (1020) prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame. This is likely to be the safest setting.</p> <p>Setting a smaller value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. While a value of 1400 may result in 6 round-trips, a value of 500 may result in 15 round-trips.</p> <p>The minimum value is 500.</p>
Return_MPPE_Keys	<p>Setting this attribute to 1 causes the module to include RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Accept response sent to the Access Point. This is necessary for the Access Point to key the WEP encryption.</p> <p>If the Access Point is authenticating only end-users and WEP is not being used, this attribute may be set to 0.</p> <p>The default value is 1.</p>
DH_Prime_Bits	<p>This attribute selects the size prime that the module uses for Diffie-Hellman modular exponentiation. The larger the prime, the less susceptible the system is to certain types of attacks. The smaller the prime, the cheaper (in CPU terms) the Diffie-Hellman key agreement operation. Supported values are 768, 1024, 1536, 2048, 3072 and 4096.</p> <p>The default value is 1024.</p>

[Server_Settings] field	Meaning
Cipher_Suites	Specifies the TLS cipher suites (in order of preference) that the server is to use. These cipher suites are documented in RFC 2246, "The TLS Protocol Version 1," and other TLS-related RFCs and draft RFCs. Default is: 0x16, 0x13, 0x66, 0x15, 0x12, 0x0a, 0x05, 0x04, 0x07, 0x09
PEAP_Min_Version (peapauth.aut only)	Specifies the minimum version of the PEAP protocol that the server should negotiate: 0 – Negotiate version 0, which is compatible with Microsoft's initial PEAP implementation (shipped in Microsoft XP Service Pack 1), 1 – Negotiate version 1, which is compatible with Cisco's initial PEAP implementation (shipped in Cisco ACU). Default value is 0.
PEAP_Max_Version (peapauth.aut only)	Specifies the maximum version of the PEAP protocol that the server should negotiate: 0 – Negotiate version 0, which is compatible with Microsoft's initial PEAP implementation (shipped in Microsoft XP Service Pack 1), 1 – Negotiate version 1, which is compatible with Cisco's initial PEAP implementation (shipped in Cisco ACU). Default value is 1.

Inner_Authentication Section

The [Inner_Authentication] section allows you to specify the way in which the inner authentication step is to operate. It consists of the following field:

[Inner_Authentication] fields	Meaning
Directed_Realm	Omitting this setting causes the inner authentication request to be handled just like any other request that has been received from a NAS. Specifying the name of a directed realm causes the request to be routed based on the methods listed in the directed realm. The default is to process the inner authentication through standard request processing.

Important: All of the filters named in these settings must be defined in the *filter.ini* file.

Request Filters Section

Request filters affect the attributes of inner authentication requests. This section consists of the following fields:

[Request_Filters] field	Meaning
Transfer_Outer_Attribs_to_New	<p>This filter affects only a new inner authentication request (rather than continuations of previous requests).</p> <p>If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.</p> <p>If this filter is not specified, no attributes from the outer request are transferred to the inner request.</p>
Transfer_Outer_Attribs_to_Continue	<p>This filter affects only a continued inner authentication request (rather than the first inner authentication request).</p> <p>If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.</p> <p>If this filter is not specified, no attributes from the outer request are transferred to the inner request.</p>
Edit_New	<p>This filter affects only a new inner authentication request (rather than continuations of previous requests).</p> <p>If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (see Transfer_Outer_Attribs_To_New, above) and attributes included in the inner authentication request sent through the tunnel by the client.</p> <p>If this filter is not specified, the request remains unaltered.</p>
Edit_Continue	<p>This filter affects only a continued inner authentication request (rather than a new inner authentication request).</p> <p>If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (see Transfer_Outer_Attribs_To_Continue, above) and attributes included in the inner authentication request sent through the tunnel by the client.</p> <p>If this filter is not specified, the request remains unaltered.</p>

Important: All of the filters named in these settings must be defined in the *filter.ini* file.

Response Filters Section

Response filters affect the attributes in the responses returned to authentication requests. This section consists of the following fields:

[Response_Filters]	
field	Meaning
Transfer_Inner_Attribs_To_Accept	<p>This filter affects only an outer Access-Accept response that is sent back to a NAS or AP.</p> <p>If this filter is specified, the filter is applied to the inner authentication response and all resulting attributes are transferred to the outer authentication response.</p> <p>If this filter is not specified, no inner authentication response attributes are transferred to the outer authentication response.</p>
Transfer_Inner_Attribs_To_Reject	<p>This filter affects only an outer Access-Reject response that is sent back to a NAS or AP.</p> <p>If this filter is specified, the filter is applied to the inner authentication response and all resulting attributes are transferred to the outer authentication response.</p> <p>If this filter is not specified, no inner authentication response attributes are transferred to the outer authentication response.</p>

***Important:** All of the filters named in these settings must be defined in the filter.ini file.*

Session _Resumption Settings

The [Session_Resumption] section allows you specify whether session resumption is allowed and under what conditions session resumption is performed. The [Session_Resumption] section consists of the following fields:

[Session_Resumption]	
field	Meaning
Session_Timeout	<p>Set this attribute to the maximum number of seconds you want the client to remain connected to the NAS or Access Point before having to re-authenticate.</p> <p>If not set to 0, the lesser of this value and the remaining resumption limit (see description below) is sent in a Session-Limit attribute to the NAS or AP on the RADIUS Access Accept response.</p> <p>If set to 0, no Session-Limit attribute is generated by the plug-in. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.</p>

[Session_Resumption] field	Meaning
Termination_Action	<p>The default value is 0.</p> <p>Setting of a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. The resumption limit can be configured to make most re-authentications fast and computationally cheap.</p> <p>Set this attribute to the integer value that you want returned in a Termination-Action attribute. This is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached.</p> <p>If you do not specify a value for this attribute, the plug-in does not generate such an attribute. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.</p> <p>The default is to not send this attribute.</p>
Resumption_Limit	<p>Set this attribute to the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature.</p> <p>This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and may not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.</p> <p>The default value is 0.</p>

Sample ttlsauth.aut File

Note: A sample peapauth.aut file would look identical to the sample ttlsauth.aut file below, apart from the [Bootstrap] section.

```
[Bootstrap]
LibraryName=ttlsauth.dll
Enable=1
InitializationString=EAP-TTLS

; Maximum TLS Message fragment length EAP-TLS will handle.
TLS_Message_Fragment_Length = 1020

; Indicates whether the EAP-TLS module should return the
; MS-MPPE-Send-Key and MS-MPPE-Recv-Key attribute upon successful
; authentication of user.
Return_MPPE_Keys = 1

; Size of the prime to use for DH modular exponentiation.
DH_Prime_Bits = 1536
```

```

; TLS cipher suites (in order of preference) that the server is to
; use.
Cipher_Suites = 0x16, 0x13, 0x66, 0x15, 0x12, 0x0a, 0x05, 0x04, 0x07,
0x09

[Inner_Authentication]
; Specifies how inner authentication routing operates.
Directed_Realm = ttls_realm

[Request_Filters]
Transfer_Outer_Attribs_to_New = My_Xfer_Out_New_Filter
Transfer_Outer_Attribs_to_Continue = My_Xfer_Out_Con_Filter
Edit_New = My_Edit_New_Filter
Edit_Continue = My_Continue_Filter

[Response_Filters]
Transfer_Inner_Attribs_To_Accept = My_Xfer_Acc_Filter
Transfer_Inner_Attribs_To_Reject = My_Xfer_Rej_Filter

[Session_Resumption]
; Maximum length of time (in seconds) the NAS/AP will allow
; the session to persist before the client is asked
; to re-authenticate.
Session_Timeout = 600

; Value to return for the Termination-Action attribute sent
; sent in an accepted client.
Termination_Action = 0

; Maximum length of time (in seconds) during which an authentication
; request that seeks to resume a previous TLS session will be
; considered acceptable.
Resumption_Limit = 3600

```

In order for this to work, you must also provide the following settings in the [ttlsauth] section of the eap.ini file:

```

First-Handle-Via-Auto-EAP = 0
EAP-Type = TTLS

```

This configuration file is re-read each time the Steel-Belted Radius server gets a HUP signal.

Examples of EAP Usage

This section contains examples of the way in which EAP can be configured in Steel-Belted Radius.

LEAP

Let us assume that you already use Steel-Belted Radius but are now planning to extend its application by deploying Wireless LANs within your company's network.

Prior to deployment of the WLAN technology, Steel-Belted Radius was being used to authenticate all end-user remote access (via a VPN gateway) and all firewall traversals. User credentials from the VPN gateway used MS-CHAP-v1 while the credentials originating from the firewall used PAP. Users are expected to provide NT Domain passwords to be granted access through Steel-Belted Radius.

In this example, the software being deployed for WLANs on all clients supports LEAP. Users are once again expected to provide their NT Domain passwords to their WLAN client software.

The administrator must perform the following steps to enable this configuration:

- 1 Nothing needs to be done to load LEAP support.

LEAP is implemented as an automatic EAP helper that is part of the core Steel-Belted Radius server.

- 2 Edit the `eap.ini` file to specify use of LEAP with the NT Domain authentication method.

The following configuration needs to appear in the `eap.ini` file:

```
[NT-Domain]
EAP-Only = 0
EAP-Type = LEAP
First-Handle-Via-Auto-EAP = 1
```

The `First-Handle-Via-Auto-EAP` setting may be 0 or 1 in this case with no significant difference in the ultimate outcome.

- 3 Re-start the Steel-Belted Radius server.

Restarting the server causes the `eap.ini` file to be re-read.

When authentication requests containing MS-CHAP-v1 or PAP credentials are received, Steel-Belted Radius passes the request to the same authentication method (NT Domain) as before.

When the request does contain EAP user credentials, the automatic EAP helper method that supports LEAP receives the request, performs the required operations (this requires multiple round-trips) and, half-way through these operations, calls the NT Domain authentication with MS-CHAP-v1 user credentials. If the NT Domain authentication succeeds, the LEAP method continues (in the remaining phase, the Steel-Belted Radius server must authenticate itself to the LEAP software running on the client).

When the LEAP automatic EAP helper successfully authenticates a user, it creates one `MS-MPPE-Send-Key` and one `MS-MPPE-Recv-Key` attribute in the final Accept response to transmit keying material back to the NAS (or Access Point) that sent it the authentication request.

LEAP and EAP MD5-Challenge

In this example, we assume that you are planning to deploy a WLAN. The software being deployed for WLANs on some of the clients supports EAP MD5-Challenge (with static WEP keys) while on other clients, LEAP is the only supported protocol. All user accounts and passwords are stored in a SQL database, with the passwords having been stored in clear text or reversibly encrypted form. Users must provide a correct user name and password to be admitted to the WLAN.

The administrator must perform the following steps to enable this configuration:

- 1 Enable loading of the SQL authentication module.
This is done by editing the `sqlauth.aut` file in the Steel-Belted Radius server directory to reflect your SQL setup and by changing the value of `Enable` in the [Bootstrap] section to 1.
- 2 Nothing needs to be done to load LEAP support.
LEAP is implemented as an automatic EAP helper.
- 3 Nothing needs to be done to load EAP MD5-Challenge support.
EAP MD5-Challenge is implemented as an automatic EAP helper.
- 4 Edit the `eap.ini` file to specify use of LEAP and EAP MD5-Challenge with the SQL authentication method.

The following configuration needs to appear in the `eap.ini` file:

```
[sqlauth]
EAP-Only = 1
EAP-Type = LEAP, MD5-Challenge
First-Handle-Via-Auto-EAP = 1
```

The `EAP-Only` setting may be 0 or 1 in this case. Set it to 0 if you also want to support traditional (non-EAP) authentication via the SQL authentication method.

If more than half of your clients support LEAP, you should make it the first EAP type in the list. Otherwise, you may want to make EAP MD5-Challenge the first EAP type in the list. Any clients for whom the EAP protocol selected by Steel-Belted Radius proves incorrect sends back an EAP-NAK requesting a different EAP protocol.

- 5 Re-start the Steel-Belted Radius server.

Restarting the server causes the SQL authentication module to be loaded and the `eap.ini` file to be re-read.

- 6 In the Configuration dialog of the administration GUI, ensure that the SQL authentication method is activated.

In Steel-Belted Radius, authentication methods can be enabled (loaded) without being activated (in use).

Assuming that LEAP was ordered ahead of EAP MD5-Challenge, when a request does contain EAP user credentials, the automatic EAP helper that implements LEAP gets to handle the request first. It sends the client an invitation to begin a LEAP sequence. If the client supports LEAP, the exchange proceeds and (after several round-trips) the LEAP module generates MS-CHAP-v1 user credentials for the user. These user credentials are then passed to the SQL authentication module, which determines its ultimate fate.

Should it succeed, the LEAP module generates one `MS-MPPE-Send-Key` and one `MS-MPPE-Recv-Key` attribute in the final Accept response to transmit keying material back to the NAS (or Access Point) that sent it the authentication request.

If the client supports EAP MD5-Challenge, it responds to the invitation to begin an LEAP exchange with an EAP-NAK indicating that it would prefer to use EAP MD5-Challenge. Steel-Belted Radius looks for the first authentication method in its list that identifies EAP MD5-Challenge as a supported protocol.

The SQL authentication method is also configured for use with EAP MD5-Challenge. Since `First-Handle-Via-Auto-EAP` is set to 1, the request is first passed to the automatic EAP helper that implements EAP MD5-Challenge. It performs the required operations (this requires two round-trips) and generates CHAP user credentials for the user. These user credentials are then passed to the SQL authentication module, which again determines its ultimate fate.

For clients that are authenticated via EAP MD5-Challenge, keying material is transmitted back to the NAS only if a profile containing the `MS-MPPE-Send-Key` and `MS-MPPE-Recv-Key` attributes is specified by the SQL authentication method.

EAP-TTLS and EAP-TLS

In this example, we assume that a customer is looking to deploy Steel-Belted Radius for the sole purpose of supporting a new Wireless LAN deployment within the company's enterprise network.

In this example, the software being deployed for WLANs on some of the clients supports EAP-TTLS while on other clients, EAP-TLS is the only supported protocol. The users of EAP-TTLS clients are expected to enter their NT Domain user password to gain access to the WLAN, while the EAP-TLS clients must have

been configured with a user certificate, matching private key and a list of root certificates from which server certificates must derive.

The customer has already deployed a PKI and end-users have already been supplied with certificates independent of the latest WLAN deployment, though only a subset of users that have certificates are to be allowed access to the WLAN. The identities of the users that are to be allowed access via the WLAN are stored in a SQL database.

The administrator must perform the following steps to enable this configuration:

- 1 Enable loading of the EAP-TTLS module.

This is done by editing the `ttlsauth.aut` file in the Steel-Belted Radius server directory and changing the value of `Enable` in the [Bootstrap] section to 1.

- 2 Enable loading of the EAP-TLS module.

This is done by editing the `tlsauth.eap` file in the Steel-Belted Radius server directory and changing the value of `Enable` in the [Bootstrap] section to 1.

- 3 Enable loading of the SQL authentication module.

This is done by editing the `sqlauth.aut` file in the Steel-Belted Radius server directory and changing the value of `Enable` in the [Bootstrap] section to 1.

- 4 Edit the `eap.ini` file to specify use of EAP-TLS with the SQL authentication method and use of no EAP protocols with NT Domain authentication.

The following configuration needs to appear in the `eap.ini` file:

```
[NT-Domain]
EAP-Only = 0

[sqlauth]
EAP-Only = 1
EAP-Type = TLS
First-Handle-Via-Auto-EAP = 1
```

The `First-Handle-Via-Auto-EAP` setting must be set to 1 for the EAP-TLS to operate properly.

- 5 Re-start the Steel-Belted Radius server.

Restarting the server causes the EAP-TTLS and EAP-TLS modules to be loaded and the `eap.ini` file to be re-read.

- 6 In the Configuration dialog of the administration GUI, ensure that EAP-TTLS and SQL authentication is activated.

In Steel-Belted Radius, authentication methods can be enabled (loaded) without being activated (in use).

7 Also make sure the NT Domain authentication is activated.

This method performs the authentications for the inner authentication requests generated by EAP-TTLS.

8 Set the order of authentication method in the Configuration dialog.

If more than half of your clients support EAP-TTLS, you should make it the first authentication method in the list. If more than half your clients support EAP-TLS, you should make SQL authentication the first method in your list. Any clients for whom the EAP protocol selected by Steel-Belted Radius proves incorrect send back an EAP-NAK requesting a different EAP protocol. The placement of NT Domain authentication in the list does not affect overall operation.

Assuming that EAP-TTLS was ordered ahead of SQL authentication, when the request does contain EAP user credentials, the EAP-TTLS module gets to handle the request first. It sends the client an invitation to begin an EAP-TTLS handshake. If the client supports EAP-TTLS, the handshake proceeds and (after several round-trips) a new authentication request results from the inner authentication sequence.

Assuming this request contains MS-CHAP-v1 credentials and EAP-TTLS is set up to perform standard routing, the new request is passed to the NT Domain authentication method, which determines its ultimate fate. Should it succeed, the EAP-TTLS module may (based on its configuration information) generate one `MS-MPPE-Send-Key` and one `MS-MPPE-Recv-Key` attribute in the final Accept response to transmit keying material back to the NAS (or Access Point) that sent it the authentication request.

If the client supports only EAP-TLS, it responds to the invitation to begin an EAP-TTLS handshake with an EAP-NAK, indicating that it would prefer to use EAP-TLS. Steel-Belted Radius looks for the first authentication method in its list that identifies EAP-TLS as a supported protocol. The SQL authentication method is configured for use with EAP-TLS, but since `First-Handle-Via-Auto-EAP` is set to 1, the request is first passed to the automatic EAP helper that implements EAP-TLS.

The helper performs the EAP-TLS handshake (this requires multiple round-trips) and, upon successful conclusion of the handshake retrieves a specific piece of information from the client certificate presented to it during the handshake (this can be the `Principal-Name` or the least-significant CN portion of the `Subject-Name` attribute of the certificate). The EAP-TLS module then passes a secondary authorization check back to Steel-Belted Radius, which asks the SQL authentication method to verify that the user exists in the SQL database. Should the SQL authentication concur, the authentication request is considered successful.

The EAP-TLS module may (based on its configuration information) generate one `MS-MPPE-Send-Key` and one `MS-MPPE-Recv-Key` attribute in the final Accept

response to transmit keying material back to the NAS (or Access Point) that sent it the authentication request.

EAP-PEAP

In this example, we assume that you already use Steel-Belted Radius but are now planning to extend its application by deploying Wireless LANs within your company's enterprise network.

Let us say that prior to deployment of the WLAN technology, Steel-Belted Radius was being used to authenticate all end-user remote access (via a VPN gateway) and all firewall traversals. User credentials from the VPN gateway used MS-CHAP-v1 while the credentials originating from the firewall used PAP. Users are expected to provide NT Domain passwords to be granted access through Steel-Belted Radius.

The software being deployed for WLANs on all clients supports EAP-PEAP. The type of user credentials used in the EAP-PEAP inner authentication is EAP-MS-CHAP-v2. Users are expected to provide their NT Domain passwords as their passwords to their WLAN client software.

The administrator must perform the following steps to enable this configuration:

- 1 Enable loading of the EAP-PEAP module.
- 2 This is done by editing the `peapauth.aut` file in the Steel-Belted Radius server directory and changing the value of `Enable` in the [Bootstrap] section to 1.
- 3 Restart the Steel-Belted Radius server.
- 4 Restarting the server causes the EAP-PEAP module to be loaded.
- 5 In the Configuration dialog of the administration GUI, ensure that EAP-PEAP is activated.
- 6 In Steel-Belted Radius, authentication methods can be enabled (loaded) without being activated (in use).
- 7 Move EAP-PEAP to the top of the list of authentication methods. (While this is, strictly speaking, not necessary, it is a bit more efficient than leaving the NT Domain authentication method at the top of the list.)

When authentication requests containing MS-CHAP-v1 or PAP credentials are received, Steel-Belted Radius skips the EAP-PEAP method (since it is marked with `EAP-Type=PEAP` and `EAP-Only=1`) and passes the request to the same authentication method (NT Domain) as before.

When the request does contain EAP user credentials, the EAP-PEAP method receives the request, performs the TLS handshake (this requires multiple round-trips) and, upon completion of the handshake, creates a new request

containing EAP-MS-CHAP-v2 user credentials. This new request skips past the EAP-PEAP method (no new request created by the EAP-PEAP method is passed back to it) and is authenticated by the NT Domain authentication method.

At the conclusion of a successful inner authentication, the default configuration for EAP-PEAP creates one `MS-MPPE-Send-Key` and one `MS-MPPE-Recv-Key` attribute in the final Accept response to transmit keying material back to the NAS (or Access Point) that sent it the authentication request.

9

- Introduction
- SNMP Traps and Alarms
- Counter Statistics
- Rate Statistics

Introduction

SNMP (Simple Network Management Protocol) is an IETF standard for communication between a central monitoring station and various devices or services on the network. Currently, SNMP is supported only on the Solaris version of Steel-Belted Radius.

SNMP communicates across various specific types, makes, and models of devices. Each device in such a network must simply be capable of receiving SNMP queries from the central station, and of responding to these queries with status information in SNMP format. Then the device is said to support SNMP. All of the devices and services that support SNMP can be configured to report to the same central station, where an SNMP-compatible graphical user interface can be used to view and interpret the data.

Any SNMPv1-compliant Manager can be used with Steel-Belted Radius. Steel-Belted Radius supports the official SNMP MIBs for RADIUS server authentication and accounting, as well as Funk Software's own custom traps and alarms.

SNMP Management Information Base (MIB)

The information that is reported via SNMP is called a *Management Information Base* (MIB). The Steel-Belted Radius MIBs are "enterprise-specific"; that is, they define SNMP information that is exclusive to Steel-Belted Radius. If you would like to view the MIBs used by the Steel-Belted Radius Sub Agent, you can find them in the `snmp` subdirectory under the server directory (usually `/radius/snmp`).

The SNMP MIB file names are:

- `rfc2271.mib` — RFC SNMP framework definitions
- `rfc2618.mib` — RFC authentication client statistics
- `rfc2619.mib` — RFC authentication server statistics
- `rfc2620.mib` — RFC accounting client statistics
- `rfc2621.mib` — RFC accounting server statistics
- `fnkrate.mib` — FUNK rate statistics
- `fnkradtr.mib` — FUNK trap definitions

SNMP Manager and SNMP Agent

The primary components in the SNMP architecture are the Manager and the Agent. These components are distributed throughout the network as follows: Any device on the network can run an SNMP Agent program. Usually one device (though there may be more than one) runs an SNMP Manager program. The Manager can be configured to retrieve information from any of the Agents on the network. This configuration depends on the supported features in the Manager program.

You typically use the Manager at a central location, and Agents on remote equipment (including RADIUS servers) you want to monitor. The Manager can be configured to poll the Agents at specific times and intervals. The result is a collection of data about devices throughout the network. A network administrator can filter the Manager's collection of data through an SNMP-compliant user interface to view and analyze the data.

For your Agent component, you must use the Solstice Enterprise Agent (SEA). This software is free of charge and is available for downloading from the Sun Microsystems web site, <http://www.sun.com>.

SNMP Sub Agent

On the individual device, the SNMP Agent role may be subdivided into a Master Agent and several Sub Agents, as follows: Any service that runs on a device may provide a Sub Agent program which is designed to report to a Master Agent program. All of the Sub Agents on a device feed their SNMP data to the Master Agent, so that the Master Agent can relay the full set of data across the network to the SNMP Manager. The Master Agent "represents" the device to the Manager. The Manager knows nothing about the Sub Agents. From the Manager's point of view, the Master Agent is the only Agent on the device.

Steel-Belted Radius supports SNMP by providing a Sub Agent daemon called `radsnmp`. The Steel-Belted Radius Sub Agent is designed to work with the Solstice Enterprise Agent (SEA) Master Agent, a daemon called `snmpdx`.

Steel-Belted Radius SNMP Sub Agent

Note: The following discussion assumes that you have configured Steel-Belted Radius for SNMP using the instructions in "Configuring SNMP Support" in the "Installation" Chapter. It also assumes that you accepted the default SNMP library and configuration file paths while running the configuration script.

When the SEA Master Agent (`snmpdx`) is started, it looks in `/etc/snmp/conf` for any Sub Agents that want to register themselves. Any file with a `.reg` (registration)

extension is checked. After you finish configuring Steel-Belted Radius for use with SNMP, `snmpdx` finds `radsnmp.reg`, which defines `radsnmp` as a Sub Agent that wants to register and load. `snmpdx`, and then looks in `radsnmp.rsrc` (resource file) for instructions on where the `radsnmp` Sub Agent resides and how to load it.

Every SNMP Sub Agent is responsible for ensuring that a TCP port is established for communication between the program that is generating the statistics, and the Sub Agent that is reporting them. The Sub Agent must ensure that this connection is available when needed, and re-established whenever it goes down.

When the Steel-Belted Radius server is restarted, a file called `radsnmp.inf` is programmatically created in the server directory (the directory that contains the radius daemon). This file contains the TCP port number over which the Steel-Belted Radius Sub Agent communicates with the radius daemon to retrieve SNMP data.

When the Sub Agent (`radsnmp`) receives an SNMP query from the Master Agent (`snmpdx`), it attempts to establish a connection to Steel-Belted Radius (`radius`) on the TCP port defined in `radsnmp.inf`. Once the connection has been established, `radsnmp` sends a request for information to `radius`, which returns the full set of Steel-Belted Radius statistics. `radsnmp` converts these statistics to SNMP format, caches them, and returns the requested information to the Master Agent. The Master Agent then returns this information to the SNMP Manager system that initiated the request.

`radsnmp` retains a copy of the SNMP-formatted statistics. Upon each query from the Master Agent, `radsnmp` determines whether its current copy is “fresh enough” to return, and behaves as follows:

- If the cached data is less than 2 seconds old, `radsnmp` returns it to the Master Agent without contacting `radius`.
- If the cached data is 2 seconds old or older, `radsnmp` requests fresh statistics from `radius`, converts them to SNMP format, caches them, and returns them to the Master Agent.

SNMP Traps and Alarms

Although Agents respond to queries initiated by the Manager, they must sometimes send information without being explicitly queried. An SNMP *trap* is a condition that is reported by that Agent to the Manager. For example, an Agent can report when an important event, such as a full file system or system crash, occurs. When a Sub Agent detects a trap condition, it reports to the Master Agent, which then signals an *alarm* by sending messages to all appropriate destinations.

The SNMP traps and alarms that Steel-Belted Radius supports are defined and described in the `fnkradtr.mib` file. In the Trap Definitions section of this file, traps are divided into three types:

- **Informational** — Informational traps are sent to report important RADIUS information that is not an error or a warning, such as when the RADIUS server daemon is loaded or unloaded or when a threshold of some kind has resulted from a previous error or warning condition.
- **Warnings** — Warning traps are sent to report RADIUS behavior that indicates a problem that has occurred or may occur, such as when the RADIUS server is unable to connect to an external SQL database or when the file system is almost full. Many of these warning traps can be diluted or have configurable thresholds (details below).
- **Errors** — Error traps are sent to report RADIUS problems that have occurred, such as when the RADIUS server is unable to initialize one or more critical components on startup. Most Error traps indicate that the RADIUS server failed to start properly for some reason, such as the inability to allocate memory from the system. Most of these traps cannot be diluted.

Dilution and Threshold

Many of the traps defined in `fnkradtr.mib` (mostly the warning traps) can be diluted. Trap event dilution refers to configuring Steel-Belted Radius so that a particular trap is sent to the SNMP Manager only once for every n occurrences of the condition that generated that trap. This allows for a fine degree of control with respect to trap generation for certain warning and error conditions.

Also, some of these traps have configurable thresholds. This allows you to set the lower and upper limits of acceptable behavior, and to generate different types of traps depending on the condition. For example, you could configure Steel-Belted Radius such that if the count of available threads (for authentication and accounting) falls below 10, it may send a warning trap to report this potential problem. When the count of available threads rises above 20, send an informational trap to report that the count is back to an acceptable level.

All SNMP trap event dilutions and thresholds are configured in the `events.ini` file. This file resides in the RADIUS server directory. The `events.ini` file can be edited to adjust these settings. If SNMP traps are important to your RADIUS system, take a moment to read the `fnkradtr.mib` file and the `events.ini` file to understand all of the options that are available to you. Depending on the configuration of your RADIUS server, some SNMP traps may be important to you, and others may not apply.

When the RADIUS server is started, a file called `radsnmptrap.inf` is programmatically created in the server directory. When SNMP support is enabled,

and an event is reported by the RADIUS server, the server reads the dynamically generated UDP port number in `radsnmptrap.inf`, create a packet that describes the event, and send it to the specified port on the local machine. Upon receiving the packet, `radsnmp` parses it and sends the trap that matches the reported event. The list of hosts that receive the trap is specified in the `/etc/snmp/conf/snmpdx.acl` file.

Acting on Information

You may need to make decisions and take action based on the information communicated in MIBs to the SNMP Manager. Your choices will be influenced by three aspects of the MIB:

- The trap/alarm
- Its corresponding value
- The severity of the trap.

For details about the traps, see the `fnkradtr.mib` file in the `/snmp` subdirectory.

SNMP Access Control

When SNMP support is enabled and configured on the Steel-Belted Radius server, it's important to note that by default, both the `radsnmp.acl` and the `snmpdx.acl` files allow read-access to all of the authentication and accounting MIBs previously listed. This access is allowed for all managers from the public and private communities. SNMP MIB access control can be configured to allow access only to the communities and managers that need it by editing these `.acl` files.

For more information about SNMP access control, and SNMP information in general, visit the Sun Microsystems web site (www.sun.com), and read the SEA (Solstice Enterprise Agent) documentation.

Counter Statistics

There are two sets of statistics counters, one set for processing authentication requests and the other for processing accounting messages.

RADIUS-Authentication-Client Statistics

The SNMP definitions for the client side of the RADIUS authentication protocol are given in detail in the `rfc2618.mib` file. The following definitions are important in understanding the Request/Response statistics:

- $TotalIncomingPackets = Accepts + Rejects + Challenges + UnknownTypes$
- $SuccessfullyReceived = TotalIncomingPackets - MalformedResponses - BadAuthenticators - UnknownTypes - PacketsDropped$
- $SuccessfullyReceived = AccessRequests + PendingRequests + ClientTimeouts$
- Access-Response includes an Access-Accept, Access-Challenge or Access-Reject

RADIUS-Authentication-Server Statistics

The SNMP definitions for the server side of the RADIUS authentication protocol are given in detail in the rfc2619.mib file. The following definitions are important in understanding the SNMP Server Counters:

- $Responses = AccessAccepts + AccessRejects + AccessChallenges$
- $Pending = Requests - DupRequests - BadAuthenticators - MalformedRequests - UnknownTypes - PacketsDropped - Responses$
- $EntriesLogged = Requests - DupRequests - BadAuthenticators - MalformedRequests - UnknownTypes - PacketsDropped$

RADIUS-Accounting-Client Statistics

The SNMP definitions for the client side of the RADIUS accounting protocol are given in detail in the rfc2620.mib file. The following definitions are important in understanding the Request/Response statistics:

- $Requests = Responses + PendingRequests + ClientTimeouts$
- $SuccessfullyReceived = Responses - MalformedResponses - BadAuthenticators - UnknownTypes - PacketsDropped$

RADIUS-Accounting-Server Statistics

The SNMP definitions for the server side of the RADIUS authentication protocol are given in detail in the rfc2621.mib file. The following definitions are important in understanding the SNMP Server Counters:

- $Pending = Requests - DupRequests - BadAuthenticators - MalformedRequests - UnknownTypes - PacketsDropped - Responses$
- $EntriesLogged = Requests - DupRequests - BadAuthenticators - MalformedRequests - UnknownTypes - PacketsDropped - NoRecords$

Rate Statistics

The rate statistics variables are defined in the `fnkrate.mib` file and derived from existing counter statistics by taking time into consideration. There are three types of rate values calculated for each of these counter statistics:

- *current-rate*: the rate measured over the most recent rate interval
- *average-rate*: the rate measured since startup, or the most recent statistics reset command
- *peak-rate*: the highest rate observed since startup, or the most recent statistics reset command

The `funkSbrRatesSecondsPerInterval` read-only variable gives the duration in seconds of the interval over which the rate statistics are gathered.

Resetting Rate Statistics

Although statistics are automatically reset if you restart the server, you can request that all statistics be reset to zero without having to restart the server. How you accomplish this operation depends on your operating system:

- Under **UNIX**, issue the command (stored in the Radius directory):
kill -USR2 *pldServer*
where *pldServer* is the process id of your Steel-Belted Radius server.
- Under **Windows**, issue the command (stored in the directory `\Radius\Service`):
radusr2

LDAP Configuration Interface

10

- LDAP Configuration Interface
- LDAP Command Line Utilities
- LDAP Virtual Schema
- LDAP Command Examples
- LDIF File Examples
- Statistics Variables (LCI Only)

LDAP Configuration Interface

This chapter explains how to use public domain LDAP utilities to populate a Steel-Belted Radius server database with RAS Client, User, Proxy, Tunnel, and other entries.

The LDAP Configuration Interface (LCI) provided by Steel-Belted Radius consists of two major components:

- An LDAP server (embedded in the Steel-Belted Radius server).
- An LDAP virtual schema that permits the LDAP server to translate each LDAP request it receives into a request that can be understood by the Steel-Belted Radius database.

This chapter provides:

- An introduction to LDAP command line utilities.
- A description of the virtual schema that Steel-Belted Radius presents to the LDAP interface.
- Information about how to use LDAP utilities to configure the Steel-Belted Radius database.
- Sample LDIF files that control the execution of LDAP utilities.
- Information about how to view rate statistics variables with LCI utilities.

LDAP Command Line Utilities

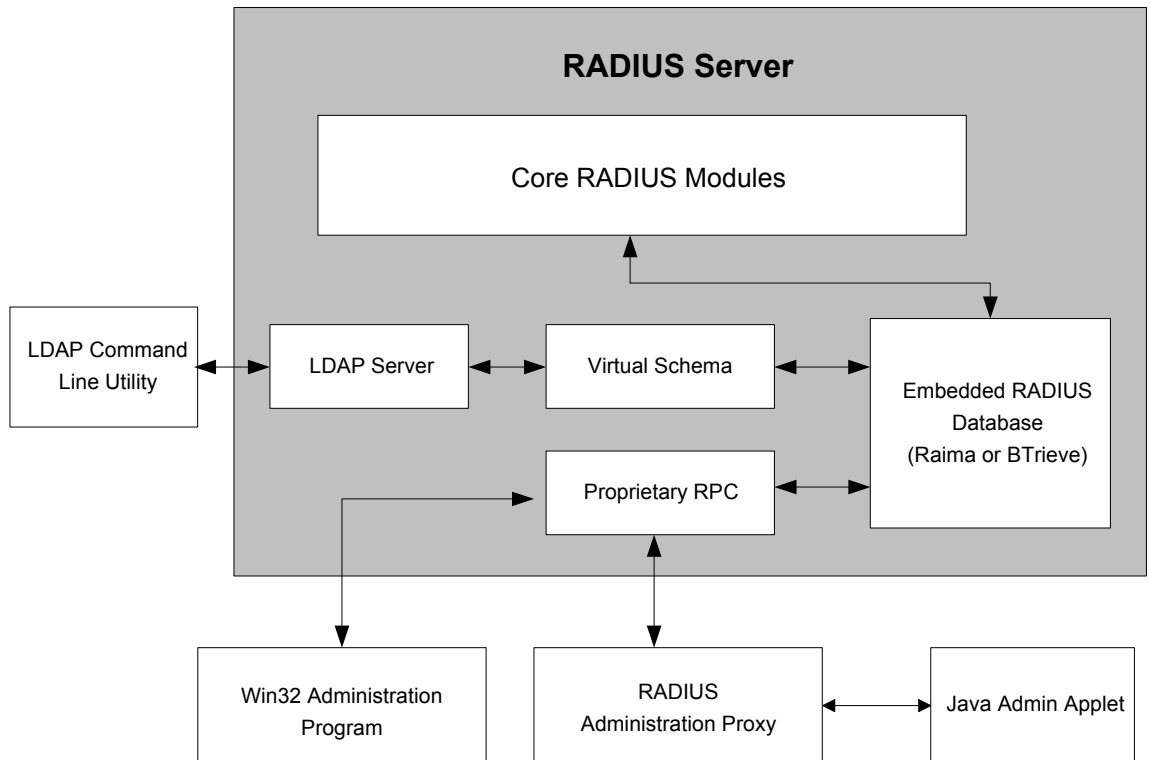
To use the LDAP configuration interface, you need the LDAP command line utilities `ldapdelete`, `ldapmodify`, and `ldapsearch`. These utilities are available through Netscape Communications as part of their free LDAP software developers' toolkit (SDK). You can download free LDAP utilities from the OpenLDAP Foundation (<http://www.OpenLDAP.org>).

The LDAP command line utilities act as clients of the LDAP server, sending requests that LDAP server processes. Requests are controlled in two ways:

- By specifying options on the LDAP command line.
- By placing instructions and data into an LDAP Data Interchange Format (LDIF) file, which you invoke on the command line by using the `-f` option.

You can think of Steel-Belted Radius as including an LDAP server and an LDAP virtual schema that translates LDAP requests into requests interpreted by the

database embedded in the RADIUS server. The following diagram illustrates the relationship between components:



See the Netscape Directory Server documentation for information about the LDIF format and the LDAP utilities.

Because communication between the LDAP client and server must occur “in the clear” (that is, not encrypted), the LDAP command line utilities should be run on the same computer as Steel-Belted Radius.

LDAP Version Compliance

The LDAP server software that has been incorporated into Steel-Belted Radius is compliant with version 2 of the LDAP specification. Therefore, we suggest using the `-V 2` command line option to direct the utilities to use version 2 features. For example:

`ldapmodify -c -V 2 -p 354 -D "cn=admin,o=radius" -w radius -f filename`

LDAP TCP Port

To avoid conflicts with LDAP services that may already be installed, the default port number for communication between Steel-Belted Radius and the LDAP client is 667. If you are certain that there will not be any conflicts, you can change this port number to 389, the standard LDAP TCP port.

You can configure Steel-Belted Radius to use a different TCP port to communicate with the LDAP client. Two steps are required. In the following example, port 354 is assigned:

- 1 In the `radius.ini` configuration file, create an `[LDAP]` section if there isn't one already, and set the `TCPPort` field to the port number you want to use. For example:

```
[LDAP]
TCPPort = 354
```

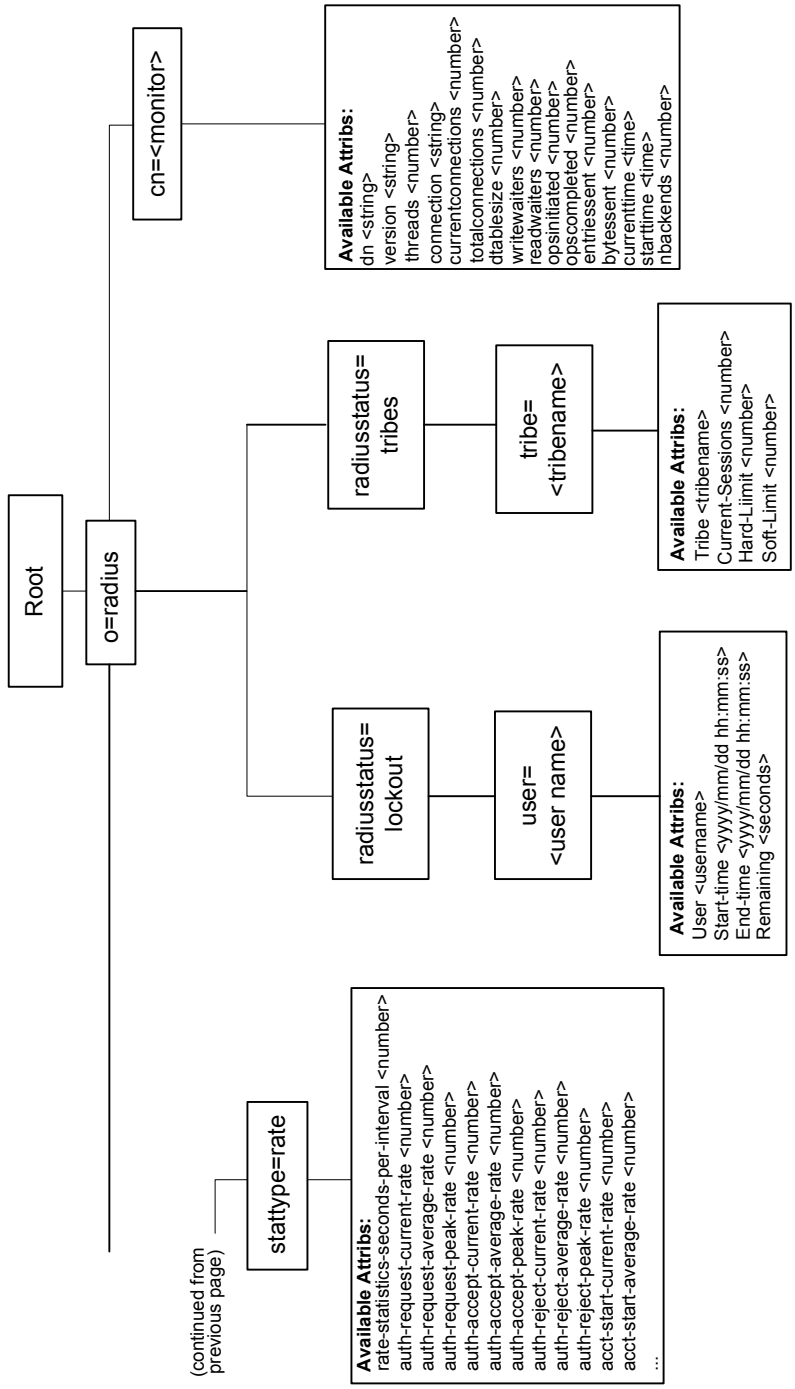
See “radius.ini [LDAP] Section” on page 218.

- 2 Specify the same port number using the `-p` option on the LDAP command line. For example:

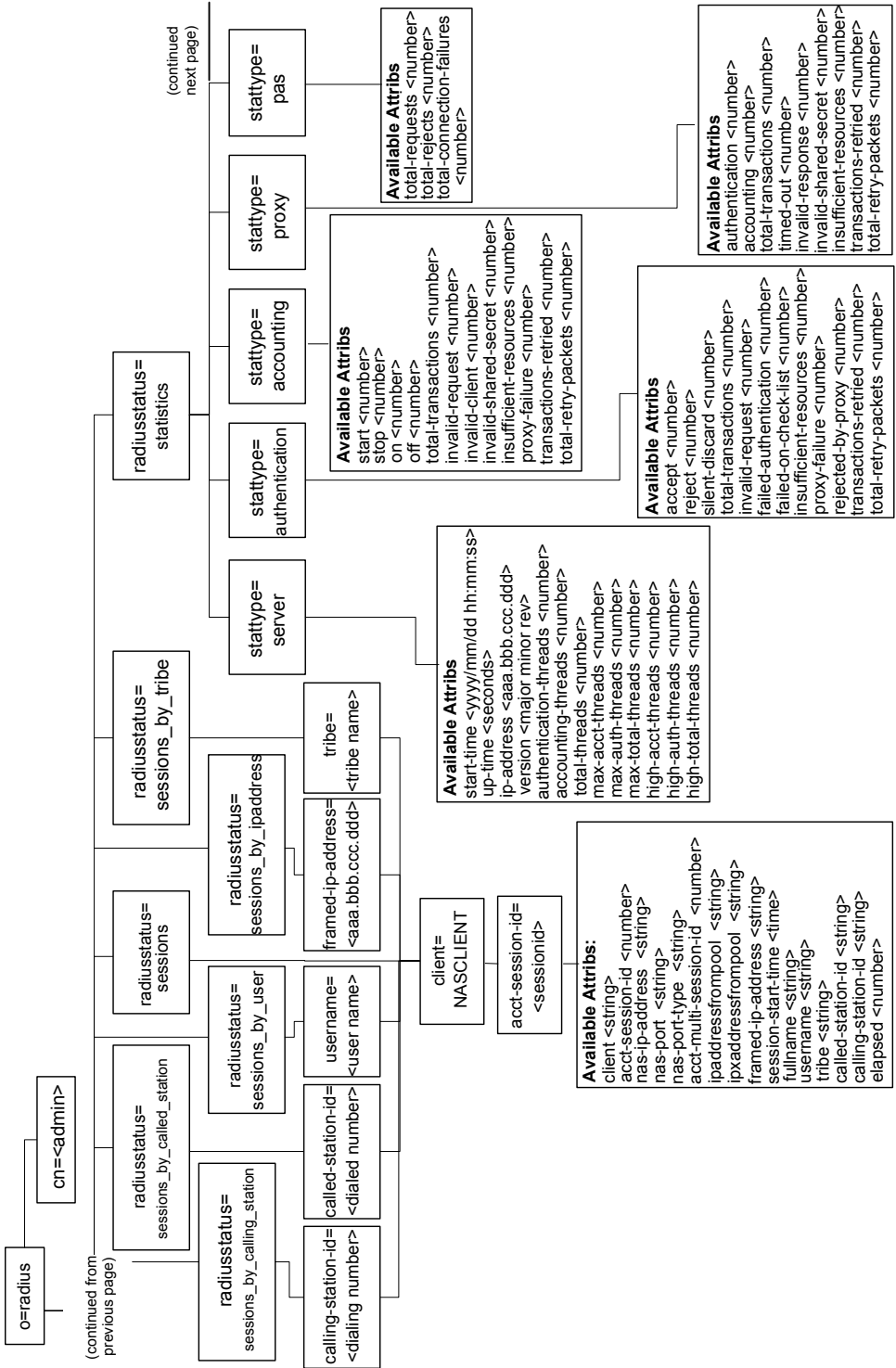
```
ldapsearch -V 2 -p 354 -D "cn=admin,o=radius" -w radius -s sub -T -b "radiusclass=Client,o=radius" radiusname=*
```

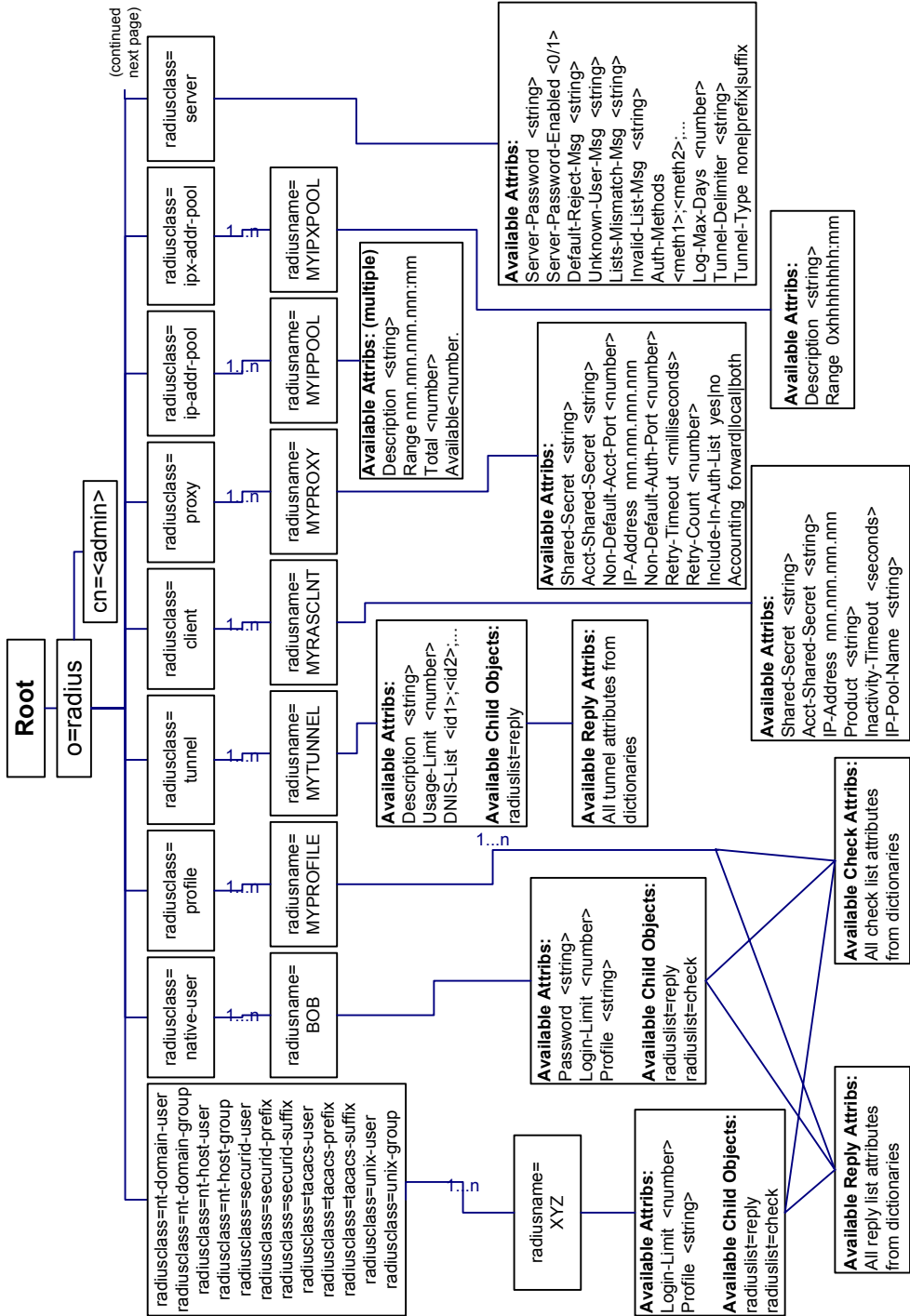
LDAP Virtual Schema

The following three figures outline the virtual schema that the LDAP server applies to configuration data so that this data can be understood by the Steel-Belted Radius database.



(continued from previous page)





Note: Your edition of Steel-Belted Radius may not support all branches of this schema.

The top-level items in the LDAP virtual schema correspond to key dialogs in the Steel-Belted Radius Administrator user interface as follows:

Item	See
radiusclass=native-user, securid-user, ...	"Users Dialog" on page 91
radiusclass=profile	"Profiles Dialog" on page 111
radiusclass=tunnel	"Tunnels Dialog" on page 118
radiusclass=client	"RAS Clients Dialog" on page 87
radiusclass=proxy	"Proxy Dialog" on page 112
radiusclass=ip-addr-pool	"IP Pools Dialog" on page 121
radiusclass=ipx-addr-pool	"IPX Pools Dialog" on page 128
radiusclass=server	"Configuration Dialog" on page 134
radiusstatus=statistics	"Statistics Dialog" on page 152
radiusstatus=sessions	"Sessions List" on page 158

While the figures show as much of the detail of the LDAP virtual schema as possible, the following rules and limitations should be considered.

Note: radiusstatus items can be read, but they cannot be modified.

Bind Request

All attempts to perform operations on the virtual schema must be preceded by an LDAP Bind request that authenticates the administrator to the Steel-Belted Radius server. The Bind request must reference a Steel-Belted Radius administrative account and must provide the password that authenticates that account. This translates into the following command line options for each invocation of the LDAP utilities:

-D "cn=AdminName,o=radius" -w AdminPassword

where **AdminName** is the administrative account name, and **AdminPassword** is its password.

Uppercase and Lowercase

The uppercase/lowercase rules for object names are the same as in the Steel-Belted Radius Administrator program; that is, almost all object names are stored in the database in uppercase format. The exception to this rule is that UNIX User/Group, SecurID User/Prefix/Suffix and TACACS+ User/Prefix/Suffix names are maintained in the case specified in the LDIF files.

Attributes

The figures do not explicitly list all the dictionary attributes that are available in the latest version of Steel-Belted Radius. The rules for entering dictionary attributes are that the attribute name must match the name found in the dictionary and the syntax type determines what is allowed for the attribute's value.

IP Addresses

The `ipaddr-pool` syntax type in the dictionary allows the user to enter an IP address or choose a pool name. If the value specified begins with the string `[pool]`, the token that follows the marker is assumed to be an IP pool name; otherwise, it must be a valid IP address. If it is neither, the operation fails.

Address ranges in IP address pool objects are specified in the form `IPAddress:NumberOfAddresses`. An example of a valid range is `128.22.12.45:34`.

IPX Addresses

The `ipxaddr-pool` syntax type in the dictionary allows the user to enter an IPX network address (up to 8 hexadecimal digits using the format `0xhhhhhhhh`) or choose a pool name. If the value specified begins with the string `[pool]`, the token that follows the marker is assumed to be an IPX pool name; otherwise, it must be a valid IPX address. If it is neither, the operation fails.

Address ranges in IPX address pool objects are specified in the form `IPXNetAddress:NumberOfAddresses`. An example of a valid range is `0xa020443b:34`.

Substrings

There are several places where a list of strings is the value of an attribute. The DNIS list in a tunnel entry and the authentication method list are two such examples. The rule for specifying the data portion for these lists is that semicolons must delimit the substrings. For example, a DNIS list for a tunnel entry might be specified as `555-1212;5551212`. If a semicolon needs to appear inside a substring, it can be “escaped” by placing a backslash character (`\`) before it.

Hexadecimal Values

Hexadecimal numbers (for attributes of syntax type `hex1`, `hex2` or `hex4`) require a `0x` prefix prior to the hexadecimal digits; for example `0x0000149a`.

Password Syntax

Passwords that are retrieved from the database may consist of either:

- A clear-text password of the form “`{x-clear}clear-text-password-string`” if the password is weakly encrypted in the database; *or*
- A string of the form “`{x-md5}xx`” if the password is stored as a one-way md5 hash; *or*
- “`{x-md5}[encrypt]clear-text-password-string`” which indicates that, although the password is specified in clear-text form, it is to be stored as a hash.

White space is treated as follows:

- When clear-text passwords are specified, the password is assumed to begin immediately following the right brace or right square bracket. Adding white space (blank, tab, and so forth) after the right brace or right square bracket causes the white space to be considered part of the password.
- White space entered at the beginning of the attribute (before the left brace or left square bracket) is ignored.
- White space entered between the right brace of `{x-md5}` and the left square bracket of `[encrypt]` is also ignored.
- All white space specified in the hexadecimal sequence describing a password hash is ignored.

Profiles, Check-Lists, and Return-Lists

Steel-Belted Radius permits user definitions to include attribute subtractions of profile entries. To signal that a user attribute is to be considered a subtraction of a profile attribute, preface the attribute value with the string `%subtract%`.

Steel-Belted Radius permits user and profile checklists to include attributes that are to be considered defaults (this allows a checklist attribute to be marked as optional). To signal that a checklist attribute is to be considered a default attribute, preface the attribute value with the string `%default%`.

Steel-Belted Radius permits user and profile Return-Lists to include attributes whose contents are to be the value of received attribute. This feature is referred to as “echoing” the attribute. To signal that a Return-List attribute is to be treated as an echo attribute, specify the attribute value as the string `%echo%`.

LDAP Command Examples

This section explains how to use the LDAP commands **ldapdelete**, **ldapmodify**, and **ldapsearch** to configure the server. Each example describes the LDAP command line options in detail.

Searching for Records

The `ldapsearch` command is used to dump information out of the LDAP tree. You can dump out information about all RAS clients with an `ldapsearch` command line similar to the following. Note there must be a blank space between each option (for example, **-p**) and its value (for example, **354**). Command syntax is case-sensitive.

```
ldapsearch -V 2 -p 354 -D "cn=oper,o=radius" -w radadmin -s sub -T -b "radiusclass=Client,o=radius" radiusname=*
```

ldapsearch Option	Meaning
-V 2	The version 2 dialect of LDAP is to be used to communicate with the server. <i>NOTE: This option is not required, but specifying it improves the performance of the transaction.</i>
-p 354	The version 2 dialect of LDAP is to be used to communicate with the server. <i>NOTE: This option is not required, but specifying it improves the performance of the transaction.</i>
-D "cn=oper,o=radius"	The command will be authenticated using an administrative account called oper. <i>NOTE: Any administrative account name may be used in place of oper in the above example. o=radius may not be changed.</i>
-w radadmin	The command is providing an authentication password of radadmin. <i>NOTE: The -w parameter value (in this case radadmin) must match the password of the account named by the -D parameter.</i>
-s sub	Recursion is to be used starting at the base.
-T	To make the output more readable, long output lines are not to be continued on the next line.
-b "radiusclass=Client,o=radius"	This is the base at which the search operation is to begin.
radiusname=*	This is the criterion which matched objects must satisfy.

If you execute the `ldapsearch` command shown above, against a Steel-Belted Radius server containing two Native User definitions, your output LDIF file may look like the following sample:

```
dn: radiusname=KEVIN,radiusclass=Native-User,o=radius
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: KEVIN
password: {x-clear}secret1
profile: ISDN
login-limit: 2

dn: radiusname=MICHAEL,radiusclass=Native-User,o=radius
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: MICHAEL
password: {x-clear}secret99
profile: ISDN
login-limit: 2
```

Modifying Records

You can modify the Steel-Belted Radius server configuration with an `ldapmodify` command line similar to the following. Note there must be a blank space between each option (for example, `-p`) and its value (for example, `354`). Command syntax is case-sensitive.

`ldapmodify -c -V2 -h hostname -p 354 -D "cn=oper,o=radius" -w radadmin -f filename`

ldapmodify Option	Meaning
<code>-c</code>	The command is to run in continuous mode; do not stop on errors.
<code>-V2</code>	The version 2 dialect of LDAP is to be used to communicate with the server. <i>NOTE: This option is not required, but specifying it improves the performance of the transaction.</i>
<code>-h hostname</code>	The name of the host to which this command applies. If none is given, the command is applied to the local database.

Idapmodify Option	Meaning
-p 354	TCP port 354 is to be used to communicate with the LDAP interface of the server. The -p value must match the TCPPort setting in the [LDAP] section of radius.ini. If the -p option is not specified, the default port number for the Steel-Belted Radius server and the LDAP utilities is used (port 389).
-D "cn=oper,o=radius"	The command is authenticated using an administrative account called <code>oper</code> . <i>NOTE: Any administrative account name may be used in place of <code>oper</code> in the above example. <code>o=radius</code> may not be changed.</i>
-w radadmin	The command is providing an authentication password of <code>radadmin</code> . <i>NOTE: The -w parameter value (in this case <code>radadmin</code>) must match the password of the account named by the -D parameter.</i>
-f filename	This is the input LDIF file to process.

*Note: You can also use the **-h** option with `ldapmodify` to specify the name of a remote host on which the LDAP interface is available. You should run the LDAP utilities remotely only if you are convinced that unauthorized snooping on the network between the LDAP client and server is not a problem.*

The difference in syntax between the LDIF files output by `ldapsearch` and those required for input to `ldapmodify` is that the `ldapmodify` input files must contain a `changetype` entry immediately following each `dn` entry in the file. The `changetype` entry specifies how to use the data to change the LDAP database.

The full syntax for `changetype` within each transaction is as follows:

```
dn: distinguished-name-of-entry
changetype: keyword
subkeyword: attribute
attribute: value
changetype: keyword
subkeyword: attribute
attribute: value
changetype: keyword
subkeyword: attribute
attribute: value
.
.
.
```

where:

keyword may be add, modify, or delete;
subkeyword may be (respectively): add, replace, or delete;
attribute may be any LDAP attribute in the entry; *and*
value is the value to assign to the attribute.

Repeated `changetype: keyword` entries are not required within a transaction unless you change the keyword. From top to bottom within the transaction, the latest keyword applies until another `changetype: keyword` entry is provided. The following syntax is perfectly valid if the same keyword applies throughout the transaction:

```
dn: distinguished-name-of-entry
changetype: keyword
subkeyword: attribute
attribute: value
subkeyword: attribute
attribute: value
subkeyword: attribute
attribute: value
.
.
.
```

`subkeyword: attribute` entries are optional and indicate that you want to apply the change to a specific attribute within the entry. If there are no `subkeyword: attribute` entries in the transaction, the change applies to the entire entry. For example, it is faster to delete an entire entry:

```
dn: radiusname=TINYCO.COM,radiusclass=Proxy,o=radius
changetype: delete
```

but if you want to delete only a few attributes from the entry, you may do so:

```
dn: radiusname=TINYCO.COM,radiusclass=Proxy,o=radius
changetype: delete
delete: retry-count
-
delete: include-in-auth-list
```

If the `subkeyword` is add or replace, an `attribute: value` entry must appear immediately following the `subkeyword: attribute` entry. If the `subkeyword` is delete, the `attribute: value` entry does not apply and should be omitted.

The following sample LDIF file could be used with an **ldapmodify** command.

```
dn: radiusname=BIGCO.COM,radiusclass=Proxy,o=radius
changetype: add
radiusname: BIGCO.COM
ip-address: 194.132.5.89
accounting: both
retry-count: 3
retry-timeout: 5000
shared-secret: testing123
include-in-auth-list: no

dn: radiusname=BIGGERCO.COM,radiusclass=Proxy,o=radius
changetype: modify
replace: shared-secret
shared-secret: hereistheseecret
-
replace: ip-address
ip-address: 192.7.2.121

dn: radiusname=TINYCO.COM,radiusclass=Proxy,o=radius
changetype: modify
delete: include-in-auth-list
```

Note: To delete the proxy entry for TINYCO.COM, issue the following command:

```
dn: radiusname=TINYCO.COM,radiusclass=Proxy,o=radius
changetype: delete
```

Adding Records

You can populate an LDAP database by creating an LDIF file that imports entries from one LDAP database into another. You can search the first database for the entries you want, then add them to the second database. You can even use the search operation to filter out attributes from the first database that you don't want in the second database. You can search the first database using `ldapsearch`. This creates an LDIF file which you can then input to `ldapmodify`.

To import entries from one LDAP database into another, start by running the `ldapsearch` command on the first database. Request only the attributes you want for the new database. When `ldapsearch` completes processing, edit the output LDIF file. After each line that begins with `dn:`, add a single line containing the text `changetype: add`. Once your editing is complete, run an **ldapmodify -f** command that references the new LDIF file. When the `ldapmodify` command finishes

processing, your new database is populated with the records you extracted from the old database.

When input to the **ldapmodify -f** command, the following sample LDIF file takes the results of our **ldapsearch** command above and adds the resulting entries to another LDAP database. The specific database is named on the **ldapmodify** command line.

```
dn: radiusname=KEVIN,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: KEVIN
password: {x-clear}secret1
profile: ISDN
login-limit: 2

dn: radiusname=MICHAEL,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: MICHAEL
password: {x-clear}secret99
profile: ISDN
login-limit: 2
```

Deleting Records

The **ldapdelete** command allows you to remove records from the LDAP database. For example, to delete entries names USER1 through USER5 enter the following information into a file called **deletexample.ldf**.

```
radiusname=USER1,radiusclass=Native-User,o=radius
radiusname=USER2,radiusclass=Native-User,o=radius
radiusname=USER3,radiusclass=Native-User,o=radius
radiusname=USER4,radiusclass=Native-User,o=radius
radiusname=USER5,radiusclass=Native-User,o=radius
```

Now, pass this file to the command as follows:

```
ldapdelete -V2 -h hostname -p 667 -D"cn=admin,o=radius" -w password -f deletexample.ldf
```

Important: Verify that the `dn:` values that usually appear in these entries are not a part of the entries in your file, because this will cause the command to fail.

You can request `ldapdelete` to remove records from the LDAP database without having to supply a file. For example, to delete the native user record identified as `USER1`, you would enter the following:

```
ldapdelete -V2 -h hostname -p 667 -D"cn=admin,o=radius" -w password "radiusname=USER1,radiusclass=Native-User,o=radius"
```

You can cause records to be deleted by means of the `ldapmodify` command, if the entries in the text file contain the line `changetype: delete`. Consider the following sample LDIF file, named `deletemodify.ldf`:

```
dn: radiusname=barry,radiusclass=Native-User,o=radius
changetype: delete
dn: radiusname=maurice,radiusclass=Native-User,o=radius
changetype: delete
dn: radiusname=robin,radiusclass=Native-User,o=radius
changetype: delete
```

This file can be passed to the `ldapmodify` command as follows:

```
ldapmodify -V2 -h hostname -p 667 "cn=admin,o=radius" -w password -f deletemodify.ldf
```

Warning: Exercise extreme caution when deleting items, as an error could actually cause the deletion of an entire container in some directory servers without any prompting for confirmation and, should that happen, the entire directory server could fail.

LDIF File Examples

This topic explains how to construct LDIF files that, when input to the `ldapmodify` command, add entries to the Steel-Belted Radius database.

For `ldapmodify` syntax, see “Modifying Records” on page 343.

We’ve provided sample LDIF entries that show you how to add database entries for RADIUS clients, Users, Proxy RADIUS targets, Tunnels, IP Pools, IPX Pools, and

RADIUS server configuration details. In general, one sample is provided per topic, but the LDIF files you construct probably contain many entries of many types.

LDIF syntax requires you to provide similar information as in the Steel-Belted Radius Administrator dialogs (a RAS Client entry requires an IP address and shared secret, a User entry requires a name and password, and so on). However, when constructing LDIF files, do not rely on your prior experience with Administrator dialogs. Knowledge of the dialogs is useful, but you must follow the database schema outlined in “LDAP Virtual Schema” on page 335.

Adding RADIUS Clients with LDIF

The following sample LDIF entry would add a RADIUS client named ANNEX105 to your Steel-Belted Radius database.

```
dn: radiusname=ANNEX105,radiusclass=Client,o=radius
changetype: add
objectclass: top
objectclass: Client
radiusname: ANNEX105
ip-address: 193.162.45.12
product: Nortel Networks Remote Annex
shared-secret: testing123
```

The syntax in this LDIF entry is as follows.

Italic text indicates values that you need to provide yourself; vertical bars ‘|’ and ellipses ‘...’ indicate that you must choose one from a set of values listed in “LDAP Virtual Schema” on page 335. Otherwise, you should enter the text exactly as shown.

```
dn: radiusname=String,radiusclass=Client,o=radius
changetype: add
objectclass: top
objectclass: Client
radiusname: String
ip-address: IPAddressOfTheClientDevice
product: Make&ModelChoiceFromVendor.IniFile | ...
shared-secret: SharedSecretThatWasConfiguredOnTheClientDevice
RASClientField: RASClientFieldValue
RASClientField: RASClientFieldValue
.
.
.
```

Adding Users with LDIF

The following sample LDIF entry would add a Native User named KEVIN to your Steel-Belted Radius database.

```
dn: radiusname=KEVIN,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: KEVIN
password: {x-clear}secret1
profile: ISDN
login-limit: 2
```

The syntax in this LDIF entry is as follows.

Italic text indicates values that you need to provide yourself; vertical bars (|) and ellipses (...) indicate that you must choose one from a set of values listed in “LDAP Virtual Schema” on page 335. Otherwise, you should enter the text exactly as shown.

```
dn: radiusname=String,radiusclass=Native-User |
    UNIX-User |..., o=radius
changetype: add
objectclass: top
objectclass: Native-User | UNIX-User | ...
objectclass: user
radiusname: String
password: {x-clear}PString | {x-md5}Hash |
    {x-md5}{encrypt}PString |...
profile: NameOfProfileEntryInTheServerDatabase
login-limit: IntegerGivingConcurrentConnectionLimit
UserField: UserFieldValue
UserField: UserFieldValue
.
.
.
```

The following sample LDIF file would add a Native User named CHRISTIAN who has various attribute/value pairs assigned to his Check-List and Return-List.

```
dn: radiusname=christian,radiusclass=native-user,o=radius
changetype: add
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: CHRISTIAN
password: {x-clear}password
login-limit: 2

dn:
radiuslist=check,radiusname=CHRISTIAN,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: check
radiuslist: check
NAS-IP-Address: 50.50.50.50
Framed-protocol: PPP

dn:
radiuslist=reply,radiusname=CHRISTIAN,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: reply
radiuslist: reply
framed-ip-address: 100.100.100.100
framed-IP-Netmask: 255.255.255.224
```

The Check-List and Return-List are objects in the LDAP virtual schema, but the individual RADIUS attributes are not. Therefore, you must use a separate LDIF entry for each the Check-List and Return-List, but each of the LDIF entries can name multiple attribute/value pairs.

To indicate that a transaction applies to the User's Check-List (rather than to the User entry itself), use the keyword `check` as the value for `radiuslist` and `objectclass` within the transaction (see syntax examples, above and below). You'll need to assign this value to `radiuslist` once in the distinguished name, and once again just before the list of attributes. You'll also need to assign the value to `objectclass`, just above the second `radiuslist` entry.

To indicate the Return-List, use the keyword `reply`.

The LDIF syntax to add a User entry, complete with a Check-List and Return-List, is as follows. Note that the `radiusname` and `radiusclass` values for all of the transactions that apply to the same User entry must be the same.

Italic text indicates values that you need to provide yourself; vertical bars (|) and ellipses (...) indicate that you must choose one from a set of values listed in “LDAP Virtual Schema” on page 335. Otherwise, you should enter the text exactly as shown.

```
dn: radiusname=String,radiusclass=Native-User | ...,o=radius
changetype: add
objectclass: top
objectclass: Native-User | UNIX-User| ...
objectclass: user
radiusname: String
password: {x-clear}PString | {x-md5}Hash | {x-md5}{encrypt}PString
|...
profile: NameOfProfileEntryInTheServerDatabase
login-limit: IntegerGivingConcurrentConnectionLimit
UserField: UserFieldValue
UserField: UserFieldValue

dn: radiuslist=check,radiusname=String,radiusclass=Native-User |
...,o=radius
changetype: add
objectclass: top
objectclass: check
radiuslist: check
AttributeName: AttributeValue
AttributeName: AttributeValue
.
.
.
dn: radiuslist=reply,radiusname=String,radiusclass=Native-User |
...,o=radius
changetype: add
objectclass: top
objectclass: reply
radiuslist: reply
AttributeName: AttributeValue
AttributeName: AttributeValue
.
.
.
```

Adding Proxy Targets with LDIF

The following sample LDIF entry would add the Proxy RADIUS target BIGCO.COM to your Steel-Belted Radius database.

```
dn: radiusname=BIGCO.COM,radiusclass=Proxy,o=radius
changetype: add
objectclass: top
objectclass: Proxy
radiusname: BIGCO.COM
ip-address: 194.132.5.89
accounting: both
retry-count: 3
retry-timeout: 5000
shared-secret: testing123
include-in-auth-list: no
```

The syntax in this LDIF entry is as follows.

Italic text indicates values that you need to provide yourself; vertical bars (|) and ellipses (...) indicate that you must choose one from a set of values listed in “LDAP Virtual Schema” on page 335. Otherwise, you should enter the text exactly as shown.

```
dn: radiusname=StringToParseAsProxyName,radiusclass=Proxy,o=radius
changetype: add
objectclass: top
objectclass: Proxy
radiusname: StringToParseAsProxyName
ip-address: IPAddressOfTheTargetServer
accounting: Both | ...
retry-count: Integer
retry-timeout: Integer
shared-secret: SharedSecretThatWasConfiguredOnTheTargetServer
include-in-auth-list: Yes | No
ProxyField: ProxyFieldValue
ProxyField: ProxyFieldValue
.
.
.
```

Adding Tunnels with LDIF

The following sample LDIF entry would add the Tunnel `ACME.COM` to your Steel-Belted Radius database.

```
dn: radiusname=ACME.COM,radiusclass=Tunnel,o=radius
changetype: add
objectclass: top
objectclass: Tunnel
radiusname: ACME.COM
dnis-list: 8005551212;6171231234;12343210
description: This is the Tunnel configuration for Acme Corp.
usage-limit: 24
```

The syntax in this LDIF entry is as follows.

Italic text indicates values that you need to provide yourself. Otherwise, you should enter the text exactly as shown.

```
dn:
radiusname=StringToParseAsTunnelName,radiusclass=Tunnel,o=radius
changetype: add
objectclass: top
objectclass: Tunnel
radiusname: StringToParseAsTunnelName
dnis-list: PhoneNumber;PhoneNumber;etc
description: StringDescribingTunnel
usage-limit: IntegerGivingConcurrentConnectionLimit
TunnelField: TunnelFieldValue
TunnelField: TunnelFieldValue
.
.
.
```

Adding IP Address Pools with LDIF

The following sample LDIF entry would add an IP address pool named `POOL1` to your Steel-Belted Radius database.

```
dn: radiusname=POOL1,radiusclass=IP-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IP-Addr-Pool
radiusname: POOL1
description: Address pool for common users
range: 198.187.100.1:50
range: 198.187.101.1:50
```

The syntax in this LDIF entry is as follows.

Italic text indicates values that you need to provide yourself; vertical bars (`()`) and ellipses (`...`) indicate that you must choose one from a set of values listed in “LDAP Virtual Schema” on page 335. Otherwise, you should enter the text exactly as shown. You may provide multiple IP address ranges using the `range` field.

```
dn: radiusname=String,radiusclass=IP-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IP-Addr-Pool
radiusname: String
description: StringDescribingPool
range: IPAddress:Range
range: IPAddress:Range
.
.
.
```

Adding IPX Address Pools with LDIF

The following sample LDIF entry would add an IPX address pool named `NETWARE1` to your Steel-Belted Radius database.

```
dn: radiusname=NETWARE1,radiusclass=IPX-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IPX-Addr-Pool
radiusname: NETWARE1
description: IPX network numbers for dial in users
range: 0xffff0a00:500
```

The syntax in this LDIF entry is as follows.

Italic text indicates values that you need to provide yourself; vertical bars (|) and ellipses (...) indicate that you must choose one from a set of values listed in “LDAP Virtual Schema” on page 335. Otherwise, you should enter the text exactly as shown. You may provide multiple IPX address ranges using the range field.

```
dn: radiusname=String,radiusclass=IPX-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IPX-Addr-Pool
radiusname: String
description: StringDescribingPool
range: IPXAddress:Range
range: IPXAddress:Range
.
.
.
```

Configuring a RADIUS Server with LDIF

The following sample LDIF entry would configure your Steel-Belted Radius server by adding the Native User authentication method and defining conventions for tunnel name parsing.

```
dn: radiusclass=Server, o=radius
changetype: add
objectclass: top
objectclass: RadiusClass
radiusclass: Server
auth-methods: Native User
tunnel-delimiter: $
tunnel-type: prefix
```

The syntax in this LDIF entry is as follows.

Italic text indicates values that you need to enter; vertical bars (|) and ellipses (...) indicate that you must choose one from a set of values listed in “LDAP Virtual Schema” on page 335. Otherwise, you should provide the text exactly as shown.

Note that there are additional configuration fields not shown in this entry; see the database schema for details.

```
dn: radiusclass=Server, o=radius
changetype: add
objectclass: top
objectclass: RadiusClass
radiusclass: Server
auth-methods: Native User | UNIX User | SecurID Prefix | ...
tunnel-delimiter: Character
tunnel-type: Prefix | Suffix | Neither
ConfigurationField: ConfigurationFieldValue
ConfigurationField: ConfigurationFieldValue
.
.
.
```

Statistics Variables (LCI Only)

There are server statistics counters to monitor the total number of certain types of events. The LCI allows you to read these statistics that monitor the performance of your Steel-Belted Radius server.

Note: See the *SNMP* chapter for the equivalent statistics supported by *SNMP*.

Counter Statistics

The statistics counters can be accessed via the LCI by executing the following one line command:

```
ldapsearch -V 2 -h 127.0.0.1 -p 667 -D "cn=admin,o=radius" -w radius -s sub -T -b "radiusstatus=statistics,o=radius" stattype=typeofstatus
```

The following sections illustrate the variables displayed for each possible setting of the **stattype** parameter.

stattype: server

```
dn: stattype=server,radiusstatus=statistics,o=radius
objectclass: top
objectclass: radiusstatus
radiusstatus: statistics
stattype: server
```

```
start-time: 2002/05/08 13:29:08
up-time: 26188
ip-address: 192.168.21.142
version: v 2.20.33
authentication-threads: 0
accounting-threads: 0
total-threads: 0
max-auth-threads: 100
max-acct-threads: 100
max-total-threads: 200
high-auth-threads: 2
high-acct-threads: 0
high-total-threads: 2
```

stattype: authentication

```
dn: stattype=authentication,radiusstatus=statistics,o=radius
objectclass: top
objectclass: radiusstatus
radiusstatus: statistics
stattype: authentication
accept: 1
reject: 0
silent-discard: 0
total-transactions: 8
invalid-request: 0
failed-authentication: 0
failed-on-check-list: 0
insufficient-resources: 0
proxy-failure: 0
rejected-by-proxy: 0
transactions-retried: 0
total-retry-packets: 0
```

stattype: accounting

```
dn: stattype=accounting,radiusstatus=statistics,o=radius
objectclass: top
objectclass: radiusstatus
radiusstatus: statistics
stattype: accounting
start: 0
stop: 0
on: 0
off: 0
total-transactions: 0
invalid-request: 0
```

```
invalid-client: 0
invalid-shared-secret: 0
insufficient-resources: 0
proxy-failure: 0
transactions-retried: 0
total-retry-packets: 0
```

stattype: proxy

```
dn: stattype=proxy,radiusstatus=statistics,o=radius
objectclass: top
objectclass: radiusstatus
radiusstatus: statistics
stattype: proxy
authentication: 0
accounting: 0
total-transactions: 0
timed-out: 0
invalid-response: 0
invalid-shared-secret: 0
insufficient-resources: 0
transactions-retried: 0
total-retry-packets: 0
```

Rate Statistics

The rate statistics variables are derived from existing counter statistics by taking time into consideration. There are three types of rate values calculated for each of these counter statistics:

- *current-rate*: the rate measured over the most recent rate interval
- *average-rate*: the rate measured since startup, or the most recent statistics reset command
- *peak-rate*: the highest rate observed since startup, or the most recent statistics reset command

Additionally, there is a (read-only) time value used in calculations:

- **Rate Statistics Seconds-per-Interval**: the duration in seconds of the interval over which the rate statistics are gathered.

To read rate statistics from the LCI, you must set **stattype: rate**. This results in output such as the following:

```
rate-statistics-seconds-per-interval: 1
auth-request-current-rate: 0
auth-request-average-rate: 0
```

```
auth-request-peak-rate: 7
auth-accept-current-rate: 0
auth-accept-average-rate: 0
auth-accept-peak-rate: 1
auth-reject-current-rate: 0
auth-reject-average-rate: 0
auth-reject-peak-rate: 0
acct-start-current-rate: 0
acct-start-average-rate: 0
acct-start-peak-rate: 0
acct-stop-current-rate: 0
acct-stop-average-rate: 0
acct-stop-peak-rate: 0
proxy-auth-request-current-rate: 0
proxy-auth-request-average-rate: 0
proxy-auth-request-peak-rate: 0
proxy-acct-request-current-rate: 0
proxy-acct-request-average-rate: 0
proxy-acct-request-peak-rate: 0
proxy-fail-timeout-current-rate: 0
proxy-fail-timeout-average-rate: 0
proxy-fail-timeout-peak-rate: 0
proxy-fail-badresp-current-rate: 0
proxy-fail-badresp-average-rate: 0
proxy-fail-badresp-peak-rate: 0
proxy-fail-badsecret-current-rate: 0
proxy-fail-badsecret-average-rate: 0
proxy-fail-badsecret-peak-rate: 0
proxy-fail-missingresr-current-rate: 0
proxy-fail-missingresr-average-rate: 0
proxy-fail-missingresr-peak-rate: 0
proxy-retries-current-rate: 0
proxy-retries-average-rate: 0
proxy-retries-peak-rate: 0
proxy-auth-rej-proxy-current-rate: 0
proxy-auth-rej-proxy-average-rate: 0
proxy-auth-rej-proxy-peak-rate: 0
proxy-acct-fail-prox-current-rate: 0
proxy-acct-fail-prox-average-rate: 0
proxy-acct-fail-prox-peak-rate: 0
proxy-auth-rej-proxy-error-current-rate: 0
proxy-auth-rej-proxy-error-average-rate: 0
proxy-auth-rej-proxy-error-peak-rate: 0
```

Resetting Rate Statistics

Although the statistics are automatically reset if you restart the server, you can also request for all statistics to be reset to zero without having to restart the server. How you accomplish this operation depends on your operating system:

- Under **UNIX**, issue the command (stored in the Radius directory):
kill -USR2 *pIdServer*
where ***pIdServer*** is the process id of your Steel-Belted Radius server.
- Under **Windows**, issue the command (stored in the directory \Radius\Service):
radusr2

SQL Authentication

11

- SQL Authentication
- SQL Authentication Process
- Configuring SQL Authentication
- Connecting to the SQL Database
- SQL Statement Construction
- SQL Authentication Header (.aut) File
- Working with Stored Procedures in Oracle

SQL Authentication

The Steel-Belted Radius server can authenticate against records stored in an external SQL database. Any attribute(s), such as username and password, can be used to query the database.

Note: SQL databases from several different vendors are supported.

External database authentication is normally used when an organization already has a large amount of user information stored in a SQL database, and this information is to be used to authenticate these users using RADIUS. Authentication against an existing database extends authentication services to user accounts without requiring an administrator to enter user information into the Steel-Belted Radius database, one entry at a time.

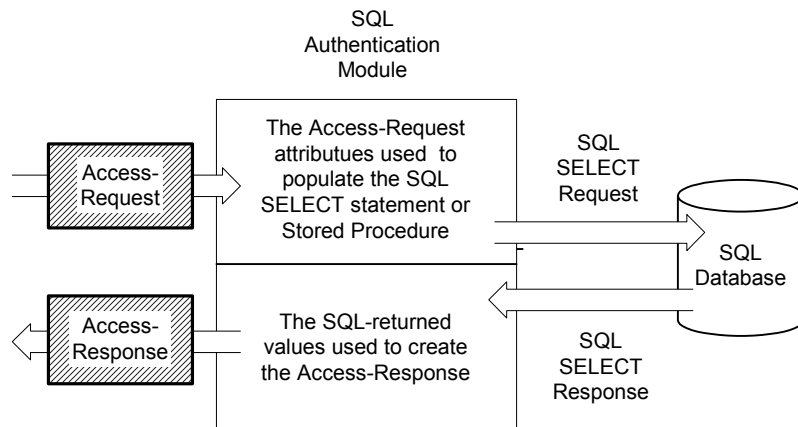
Steel-Belted Radius offers the SQL Authentication feature as a plug-in software module. Key features of the SQL plug-in include:

- The SQL statement is completely user-specified, allowing support of existing tables with existing field names and formats.
- The SQL statement supports a wide range of arithmetic and string expressions as part of the statement.
- The SQL statement is parameterized, so it is compiled once, and each execution uses variable data without need for recompilation.
- By using stored procedures, you have the ability to utilize “server-side processing” functionality which allows the SQL server to manipulate the information specified by variables.
- Multiple authentications may be overlapped at the same time.
- The SQL authentication method appears in the Configuration dialog, and may be activated and deactivated, and ordered with respect to other authentication methods.
- Multiple instances of the SQL Authentication module may operate simultaneously, allowing authentication to multiple databases.
- If the database connection drops, it is automatically reestablished after a configurable timeout, without the necessity of restarting Steel-Belted Radius.
- Data from the database can be returned in the Access-Accept as attributes.

Important: While Steel-Belted Radius does its best to provide uniformity in the operation of databases from different vendors, difference occur, particularly in the way SQL statements are interpreted. The capabilities of the SQL Authentication module depend on the capabilities of the underlying databases and their clients; things that work with one database may not work with another.

SQL Authentication Process

Any RADIUS attribute (or Steel-Belted Radius request variable) from the request can be used in a SQL SELECT statement. Any return-list attribute (i.e., a Steel-Belted Radius response variable) can be retrieved from a SQL database and returned in the Access-Response.



The SQL Authentication Process

Configuring SQL Authentication

You must configure both Steel-Belted Radius and the SQL database to support SQL authentication. The exact configuration procedure must be tailored to the database that you use. However, all procedures must give the following results:

- The required transport must be in place between SQL client software and the SQL server.
- The SQL server must be configured via a plug-in to coordinate with SQL client software.

- The Steel-Belted Radius server must be configured to communicate with the SQL client software in order to interact with the back-end SQL server to perform stored procedures or SQL queries.

Using the SQL Authentication Header File

To configure SQL Authentication, you must edit the authentication header file, `radsql.aut` (under **UNIX**) or `sqlauth.aut` (under **Windows**), located in the same directory that contains the Steel-Belted Radius service (normally `C:\RADIUS\Service`) or daemon. A reference listing of all header file options appears below. Most of these options may be left at their original settings; however, you must modify certain options to accommodate your own database.

See “SQL Authentication Header (.aut) File” on page 373.

After you complete your changes to `radsql.aut` or `sqlauth.aut` and restart Steel-Belted Radius, the `InitializationString` value that you entered in the [Bootstrap] section of this file appears in the Configuration dialog’s Authentication Methods list. You can then enable, disable, or prioritize your SQL database just like any other authentication method in the list.

See “Configuring the Authentication Sequence” on page 38.

Using Multiple SQL Databases

You can configure Steel-Belted Radius to authenticate users against more than one SQL database. Each database that you set up in this way becomes a separate selection in the Configuration dialog’s Authentication Methods list.

To add an additional database, create a new header file with extension `.aut` in the same directory as `radsql.aut` (under **UNIX**) or `sqlauth.aut` (under **Windows**). You can give this file any name you like, provided its extension is `.aut`. At startup, Steel-Belted Radius enumerates all `.aut` files to create its list of authentication methods.

When creating the new file, start by copying the original `.aut` file (`radsql.aut` (under **UNIX**) or `sqlauth.aut` (under **Windows**)). Be sure to change its `InitializationString` entry to a unique authentication method name; otherwise, Steel-Belted Radius has no way of distinguishing between the different methods in the Authentication Methods list.

Connecting to the SQL Database

Upon startup, the SQL Authentication module connects to the database, based on a connect string specified in the header file. The connect string contains information such as the name and location of the database, and the password required to connect. The connect string is passed to the database client to establish the connection.

While a sample connect string is provided in the original header file, you must configure the `Connect` entry of the header file with a connect string appropriate to your database.

It is important that the password for database access be provided as part of the connect string. If it is not:

- Under **UNIX**, the connection fails.
- Under **Windows**, at startup and each time a reconnect is required, a pop-up dialog prompts you to enter the password before making the connection.

If the initial attempt to connect to the database fails, or if in the course of processing an error occurs that the SQL Authentication module interprets as a database connection failure, the SQL Authentication module drops the connection and attempts to establish a new connection after a period of time. In the interim, all authentication requests are ignored.

The SQL Authentication module uses an exponential back-off strategy in determining how long to wait before attempting a new connection, as well as how frequently this attempt should be made. After the first dropped connection, it waits a certain amount of time before attempting to reconnect. If this attempt to reconnect also fails, it waits for twice the amount of time before trying again; and so on, up to some maximum wait time. The initial and maximum wait times are configurable.

Warning: (UNIX only): Detailed error information may not be available if there is an error processing the database logon at connect time. A numeric result code is displayed in the log. You may need to refer to product-specific documentation to decode this result code. With Oracle on UNIX, you can use the `oerr facility-code error-number` command with a facility code of `ora` from the UNIX command shell.

SQL Statement Construction

The authentication transaction is based on a SQL query that returns a password (and possibly other information) based on the name entered by the user attempting to log in.

While a sample SQL query is provided in the original header file, you must configure the SQL entry of the header file with a query appropriate to your database. The query you enter must be either a **SQL SELECT** or **SQL EXECUTE** statement that contains additional syntax elements which are preprocessed by the SQL Authentication module.

The SQL Authentication module executes SQL statements in parameterized form. This means that the SQL statement is compiled once, with parameter markers (usually question marks) as placeholders for data items that vary from one execution to the next. Only upon execution of the statement are the actual data values supplied.

The SQL statement you compose must not include parameter markers directly. Instead, the names of the parameters should be included where parameter markers would appear, in a format described below. The SQL Authentication module translates the SQL statement provided, replacing parameter names with parameter markers prior to passing the SQL statement to the database engine.

The SQL statement can be very simple. Basically, all that is required is to look up a password and possibly some optional information based on a user name. The SQL statement can also be quite complex; it can include inner joins, and it can contain expressions. The underlying database engine is responsible for handling the SQL statement; the SQL Authentication module performs no interpretation of the SQL statement other than to translate parameter names to parameter markers.

Example:

```
SELECT password, profile, fullname FROM usertable WHERE username = %name/63s
```

As shown in the example above, a parameter consists of a percent sign (%), the name of the parameter and a format specifier. The following parameter names may be used:

Item	Meaning for SQL Authentication
%OriginalUserName	The original full identification of the user, prior to any processing (i.e., user@realm).
%User	The user portion of OriginalUserName (the section before '@').
%UserName	The full user identification (user and realm strings) after all stripping and processing has been performed.

Item	Meaning for SQL Authentication
%Name	Synonym for UserName.
%EffectiveUser	The name of the user (the section before '@') as presented to the authentication method (i.e., possibly modified).
%Realm	The realm portion of the original user identification (the section after '@') as presented to the authentication method (i.e., possibly modified).
%EffectiveRealm	The realm portion of the user identification as presented to the method (i.e., possibly modified).
%NASName	The name of the NAS device, as specified in a RAS Clients entry in the Steel-Belted Radius database.
%NASAddress	The address of the NAS device, in dotted notation.
%NASModel	The make/model of the NAS device, as specified in the Steel-Belted Radius database.
%Password	The PAP password.
%AllowedAccessHours	The times that the user is allowed to be logged in.

Along with these parameters, any RADIUS attribute received in the Access-Request can be referred to by using an at-sign ('@') followed by the name of the attribute. If you need to specify a literal at-sign character in an SQL statement, such as in a User-Name, you must use two at-signs in a row. For example:

```
SELECT foo FROM bar WHERE field = 'abc@@xyz'
```

Likewise, if you need to specify a literal percent character ('%') in an SQL statement you must use a two percent characters in a row.

The format specifier should describe the database storage format of the column that corresponds to the parameter. It consists of a slash ('/'), a length, and a type, which for SQL Authentication is always 's' for string. For example, if the user's name is stored in the database as a string of up to 63 bytes, you would enter:

%name/63s

Warning: Be sure to specify a length no greater than the actual field size in the database. The compilation of the SQL statement may fail if a parameter size greater than the actual field size is specified.

Password Parameters

Normally, the only parameter you'd include in the SQL statement is %name. The %password parameter is available to support databases containing non-unique usernames. For example, your database might allow two people named "George"; one with password "swordfish", and the other with password "martha". You can authenticate them correctly with the following query:

SELECT password, profile, fullname FROM usertable WHERE username = %name/63s and password = %password/63s

You must return the password as the first column of the result to perform authentication. If the password is not returned in a password column or as an output parameter, no password authentication is performed.

In the following statement, for example, %name is an input parameter used to look up a record.

```
SELECT profile FROM database WHERE username = %name
```

Since there's no password output parameter, no password authentication is performed. The [Results] section of the .aut file should look something like the following to work with the above SELECT statement:

```
[Results]
Password=0
Profile=1/50
Alias=0
```

If the record cannot be found in the database, the authentication attempt fails.

Note: If you are not using password checking for authentication, the Password parameter must be set to 0 in the [Results] section.

Overlapped Execution of SQL Statements

The SQL Authentication module is multi-threaded. SQL Authentication can be configured with a maximum number of simultaneous executions of any SQL statement, using the `MaxConcurrent` entry in the .aut file's [Settings] section.

If `MaxConcurrent` is set to 1, SQL execution occurs serially, and the SQL execution for each authentication request must complete before execution for the next request may begin.

By increasing `MaxConcurrent`, it may be possible to increase throughput by overlapping operations, especially if the database server is remote and a large part of the time to complete a statement execution is taken up by network latency. If the database server is local, the point of diminishing returns may be reached at a small value of `MaxConcurrent`, possibly even at 1 or 2. The optimum value is a matter of experiment.

Warning: A setting of `MaxConcurrent = 1` should be sufficient for all but the most demanding environments. Increase this value only slowly and conservatively.

You might expect that databases that are licensed by number of connections would debit a single connection regardless of how many SQL statements are active. This is

not necessarily the case; some databases count each open compiled SQL statement against the licensed number of connections. So another factor that determines how `MaxConcurrent` should be set might be the database license.

The %Result Parameter

The `%result` parameter is a string value that can be returned as a column or stored procedure output parameter. The `%result` parameter can be used with or without password authentication.

The value expected to be returned in this parameter when authenticating a user can be specified in the `SuccessResult` entry of the [Settings] section. For example, if a user is successfully authenticated by the SQL Authentication method, the result signifying success is the text string “okay”. This can be automatically checked by the following setting.

```
[Settings]
SuccessResult = okay
```

Note: The string comparison is case insensitive.

If the SQL statement succeeds but the `SuccessResult` value does not match the expected value returned from the database, Steel-Belted Radius issues a reject response, which can include any attributes and values configured in the [FailedSuccessResultAttributes] section of the `*.aut` file.

In the following statement, `%password` is passed to a stored procedure, which returns a `%result` of either “okay” or something else (that signifies a rejection):

```
BEGIN CheckUser(%name, %password, %result!o); END;
```

Another example might be a database of usernames, passwords, and account status. The administrator can enable a user by setting account status to “okay”, disable by setting to some other value, without having to delete the record. In the following statement, both password and result columns are checked:

```
SELECT password, result FROM database WHERE username =
%name
```

```
[Results]
Password=1/50
%Result=2/50
Profile=0
Alias=0
```

SQL Authentication and Password Format

Steel-Belted Radius supports the authentication of users residing in a SQL database, in which password values for the users are stored in one of the following formats: clear text, UNIXcrypt, Secured Hash Algorithm (SHA1+Base64 hash), MD4 hash, or enc-md5 reversibly-encoded password.

Hashed Passwords

Values in the Password column include a prefix that indicates how the password has been processed. The prefix is in clear text between curly braces ‘{’ ‘}’ and is immediately followed by a hash value computed from the password. If no prefix is present in the value retrieved from the table Password column, the entire password is assumed to be in clear text format. In summary:

- `PasswordText` indicates clear text format (no encryption)
- `{crypt}HashHash` indicates UNIXcrypt format
- `{SHA}HashHashHash` indicates SHA1+Base64 hash encryption
- `{md4}HashHash` indicates MD4 hash of the Unicode form of password

Note: Refer to RFC 2759 for details about how MS-CHAP-V2 produces an MD4 hash value.

- `{enc-md5}EncryptedEncrypted` indicates a reversibly encrypted password

Note: Although Steel-Belted Radius reads passwords encoded in this format, you must purchase the Software Developer’s Kit to convert clear-text passwords to this format.

UNIXcrypt is the standard hash algorithm that is used for the `/etc/passwd` file on UNIX systems. This may be necessary if, for example, the standard user database on a UNIX machine (the `/etc/passwd` file) is migrated to a SQL database, so that the values in the Password column of the SQL table are processed with UNIXcrypt.

Steel-Belted Radius may be configured to expect that the values retrieved from the SQL table Password column during authentication have been run through UNIXcrypt by adding the following entry into the [Settings] section of the SQL authentication header file:

```
PasswordFormat=3
```


Automatic Parsing

If `PasswordFormat` is set to 0, Steel-Belted Radius attempts to determine the password format automatically by parsing it. This is the recommended setting. Automatic parsing expects the password to be stored in one of the formats above.

Note: The setting for automatic password parsing in older versions of Steel-Belted Radius, `auto`, has been deprecated.

SQL Authentication Header (.aut) File

The header file used to configure the SQL Authentication module must have the extension `.aut`. The format of this header file is comparable to that of a Windows INI file. It is composed of several sections; each section may contain multiple entries. Section names are enclosed in square brackets; entries are of the form `attribute=value`.

SQL Authentication [Bootstrap] Section

The [Bootstrap] section of the SQL authentication header file specifies information that Steel-Belted Radius uses to load and start the SQL Authentication module.

.aut File	
[Bootstrap] Field	Meaning for SQL Authentication
LibraryName	This entry must be set to the name of the SQL authentication module. UNIX only: The name should be <code>radsqL_auth_ora.so</code> (for Oracle) Windows only: The name should be <code>SQLAUTH.DLL</code>
Enable	This entry must contain a 1 to enable the module, 0 to disable it. If disabled, the authentication method is unavailable and does not appear in the Configuration dialog's Authentication Methods list.
InitializationString	This entry is used to specify the name of the authentication method to appear in the Configuration dialog's Authentication Methods list. In the original header file, this entry is set to <code>SQL</code> . You may alter this name if you want. The name of each authentication method must be unique. If you create additional <code>.aut</code> files to implement authentication against multiple databases, be sure that each <code>InitializationString</code> is set to a different method name.

SQL Authentication [FailedSuccessResultAttributes] Section

The [FailedSuccessResultAttributes] section of the SQL authentication header file can be used to map any RADIUS attribute returned from the database. Attributes can be specified in two ways:

- Attributes can be specified with a literal value enclosed in single quotes. Values must be enclosed with single quotes, even when they represent numeric values.
- Attributes can be specified with a numeric value that correlates to the ordering of values returned from the SQL `select` statement.

Precede attribute names with '@' and enter them as they appear in the dictionary (.dct) files. Enclose attribute values (including integers and IP addresses) in single quotes. For example:

```
[FailedSuccessResultAttributes]
@Reply-Message = 'Please re-enter your password.'
@Filter-Id = '3'
```

SQL Authentication [Failure] Section

The [Failure] section of the SQL authentication header file can be used to determine the result of the authentication process (accept or reject) when connectivity to all of the configured SQL databases has failed. For example:

```
[Failure]
Accept = 1
Profile = XYZ
FullName = Unauthenticated!
```

The following fields may be present:

Note: The Profile option and the Alias option cannot be used together. Read the descriptions below and choose the one that suits your needs.

.aut File	
[Failure] Field	Meaning
Accept	If Accept is set to 1, Steel-Belted Radius returns an Access-Accept packet with the Profile, FullName, and/or Alias attributes specified in the corresponding [Failure] section fields. If Accept is set to 0, the user is rejected.
Profile	This is the name of an existing Steel-Belted Radius Profile entry, whose Check-List and Return-List attributes are applied to the user's connection. See "Profiles" on page 58 and "Resolving profile and User Attributes" on page 59.

.aut File**[Failure] Field Meaning**

FullName	By indicating a FullName, Steel-Belted Radius returns a value in the class attribute, allowing for all [Failure] connections to be accounted.
Alias	<p>As an alternative to using the Profile parameter, you can use the Alias parameter to name an existing Steel-Belted Radius Native User entry. Steel-Belted Radius then applies the Check-List and Return-List attributes of this User entry to the user's connection.</p> <p><i>NOTE: The Alias feature permits the Concurrent connection limit (settable in the Users dialog, but not in the Profiles dialog) to be applied to the user's connection. See "Concurrent User Connections" on page 79.</i></p> <p><i>NOTE: Native User entries without passwords automatically cannot be authenticated. This is a safety feature built into Steel-Belted Radius. Therefore, setting up User entries in preparation for using the Alias parameter with SQL authentication does not pose a "back door" security risk.</i></p> <p><i>NOTE: The Native User authentication method displayed in the Configuration dialog does not need to be activated for the Alias feature to work.</i></p> <p><i>NOTE: Individual attributes retrieved from the external database override profile attributes of the same name.</i></p>

SQL Authentication [Results] Section

The [Results] section of the SQL authentication header file maps the columns named in its `SELECT` query to the type of data that Steel-Belted Radius expects these columns to contain.

The following fields may be present in a [Results] section. Each field represents a type of data required to authenticate an Access-Request, and if desired, apply authorization information as well.

Note: The Profile option and the Alias option cannot be used together. Read the descriptions below and choose the one that suits your needs.

.aut File**[Results] Field Meaning**

%LoginLimit	The name of the variable specifying the Maximum Concurrent Connection limits.
-------------	---

.aut File	[Results] Field	Meaning
%Password		<p>The value returned from this column is understood to be the user's password. The value returned by the SQL query is then matched with the user's password received in the Access-Request.</p> <p>By default, Steel-Belted Radius expects the user's password to be stored in the SQL table in clear text format. If you want to configure Steel-Belted Radius to expect that the password value is encrypted with UNIXcrypt, then in the [Settings] section of the SQL authentication header file, set PasswordFormat to 3.</p> <p>See "SQL Authentication and Password Format" on page 372.</p>
%Profile		<p>The value returned from this column is interpreted as the name of the profile to associate with the user. The value returned by the SQL query is matched with an existing Profile entry of the same name. If the value is <code>prof1</code>, and a Profile called <code>prof1</code> exists in the Steel-Belted Radius database, any Return-List or Check-List attributes in <code>prof1</code> are applied to the user's connection.</p> <p>If the value cannot be matched with an existing Profile in the Steel-Belted Radius database, the user is rejected due to "Insufficient Resources."</p>
%ProxyRealm		<p>The realm to which the authentication must be proxied. If ProxyRealm is not set, Routed Proxy does not occur.</p> <p>See "Routed Proxy" on page 461.</p>
%ProxyUserName		<p>The User-Name attribute, which must be sent in the proxy request. If ProxyUserName is not set, the User-Name from the original request packet is used.</p> <p>See "Routed Proxy" on page 461.</p>
%Alias		<p>The value returned from this column is matched with an existing Steel-Belted Radius Native User entry of the same name.</p> <p>If the value is <code>max1</code>, and a Native User called <code>max1</code> exists in the Steel-Belted Radius database, then any Return-List or Check-List attributes, as well as any concurrent connection limit configured for <code>max1</code>, are applied to the user's connection.</p> <p>If you want to apply concurrent connection limits to users who are being authenticated via SQL, you must set up a Native User entry specifically for this purpose, with no password.</p> <p>Important: You are strongly recommended to use %Profile, as use of %Alias has been deprecated. The %LoginLimit value allows you to implement the concurrent connection limits previously available through %Alias.</p> <p>NOTE: Native User entries without passwords automatically cannot be authenticated. This is a safety feature built into Steel-Belted Radius. Therefore, setting up Native User entries in preparation for using the Alias parameter with SQL authentication does not pose a "back door" security risk.</p>

.aut File

[Results] Field

Meaning

	<p>Generally, even if a very large number of users resides in the SQL database, you need to add only one or two Native User entries to the Steel-Belted Radius database. The concurrent connection limit associated with a single Native User entry may be applied to any number of users in the SQL database. Often a Native User entry with a connection limit of 1, and a second Native User entry with a connection limit of 2, is sufficient for an entire SQL database.</p> <p>For example, analog users may be allowed a connection limit of 1, while ISDN users are allowed a connection limit of 2.</p> <p><i>NOTE: The Native User authentication method displayed in the Configuration dialog does not need to be activated for the %Alias feature to work.</i></p>
%FullName	<p>The value returned from this column is interpreted as the full name of the user. This feature is often used to distinguish the user's full name from the actual User-Name sent in the Access-Request.</p>
RADIUS attributes	<p>Any RADIUS attribute (preceded by an '@') can be returned from the database and mapped into the [Results] section. Use attribute names as they appear in the .dct files.</p>

Consider the following `SELECT` statement:

```
SELECT user_pwd, attribs, fullname FROM rasusers WHERE user_id = %name/40
```

where `user_pwd`, `attribs`, `fullname` and `user_id` are the names of columns in the SQL table, and `rasusers` is the name of the SQL table itself. The [Results] section of this header file must map the SQL table columns `user_pwd`, `attribs`, and `fullname` to authentication and/or authorization data types; for example.

```
[Results]
Password=1/48
Profile=2/48
FullName=3/48
```

Columns in the SQL query are identified in the [Results] section by number; that is, 1 represents the first column in the `SELECT` query (from left to right), and if other columns are also referenced, 2 represents the second, and 3 the third.

Along with a number representing the column order, each entry in the [Results] section also specifies the storage format of the column in the SQL table, using the same slash ('/'), length, and type conventions as the SQL query.

Default [Results] Parameters

The `DefaultResults` flag in the [Settings] section of `sqlauth.aut` specifies whether default values for `Password`, `Profile`, `Alias` and `FullName` are automatically bound to the returned SQL data. The default `sqlauth.aut` file sets it to 0.

With `DefaultResults=0`, the results list is no longer automatically bound, and only explicit columns in the [Results] section, or embedded Parameters to a stored procedure, are used. This is the recommended setting.

The `DefaultResults=1` option remains only for backward-compatibility with old `.aut` files that rely on the default results behavior to ensure that the set of default columns are automatically bound.

SQL Authentication [Server] Section

Steel-Belted Radius can maintain multiple SQL server connections and authenticate users against authentication databases in a round-robin fashion. This convention distributes the authentication workload across several servers.

The [Server] section of the SQL authentication header file gives Steel-Belted Radius a pool of servers from which to create the round-robin list. The [Server] section names each server that might be used. It also provides rules for when each of the possible servers should be included in (or excluded from) the round-robin list.

The syntax is as follows:

```
[Server]
  ServerName=TargetNumber
  ServerName=TargetNumber
  .
  .
  .
```

[Server] Field	Meaning
ServerName	The name of the header 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 may be left blank.

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. An activation target of 0 indicates that, in the current configuration, this machine is never used.

SQL Authentication [Server/name] Sections

You must provide a [Server/name] section for each server you've named in the [Server] section, as follows, depending on your operating system:

- Under **UNIX**:

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

where the actual *username* and *password* is specific to the SQL database, and *servicename* is either the Oracle SID (for Oracle versions prior to 8i) or the Oracle service name.

- Under **Windows**:

```
[Server/name]
Connect=DSN=dsnname;UID=username;PWD=password
```

where the actual *dsnname*, *username*, and *password* is specific to the SQL database you are using.

Note: Do not use either the SA account or leave the password blank.

Last Resort Server

You may 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; in the UNIX example, 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).

- Under **UNIX**

```
[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
```

- **Under Windows**

```
[Server]
s1=2
s2=2
s3=1
[Server/s1]
Connect=DSN=dsnname;UID=username;PWD=password

[Server/s2]
Connect=DSN=dsnname;UID=username;PWD=password

[Server/s3]
Connect=DSN=dsnname;UID=username;PWD=password
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 Authentication [Settings] Section

The [Settings] section of the SQL authentication header file defines parameters that control the database connection.

.aut File	
[Settings] Field	Meaning for SQL Authentication
ConcurrentTimeout	Specifies the number of seconds a request may wait for execution before it is discarded. Since there may be only up to <code>MaxConcurrent</code> SQL statements executing at one time, new requests must be queued as they arrive until other statements are processed.

.aut File	
[Settings] Field	Meaning for SQL Authentication
Connect	<p>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.</p> <p>The exact format of the connect string depends on the database you use: see the configuration instruction file in the same directory that contains the Steel-Belted Radius service or daemon.</p>
ConnectTimeout	<p>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 may or may not implement the feature.</p>
DefaultResults	<p>If set to 0 (as now prescribed in Steel-Belted Radius) no default values are assumed and the user must explicitly enter all result items (if you are not calling a stored procedure).</p> <p>If set to 1, the default values for Results are used. This is the backward-compatibility setting and the setting if no value is specified in the file. In this case, each Result item must be explicitly specified.</p> <p>See “SQL Authentication [Results] Section” on page 375.</p>
LogLevel	<p>Activates logging for the SQL authentication component and sets the rate at which it writes entries to the server activity log file (.LOG). The LogLevel may be the number 0, 1, or 2, where 0 is the lowest logging level, 1 is intermediate, and 2 is the most verbose. 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. The LogLevel is re-read whenever the server receives a HUP signal.</p> <p>See “radius.ini [Configuration] Section” on page 212.</p> <p>See also “Radius Log File” on page 144.</p>
MaxConcurrent	<p>Specifies the maximum number of instances of a single SQL statement that may be executing at one time.</p>
MaxWaitReconnect	<p>Specifies the maximum number of seconds to wait after successive failures to reconnect after a failure of the database connection.</p> <p>WaitReconnect specifies the time to wait after failure of the database connection. This value is doubled on each failed attempt to reconnect, up to a maximum of MaxWaitReconnect.</p>
ParameterMarker	<p>This is the character or sequence of characters used as the parameter marker in a parameterized SQL query. Normally, this is the question mark ('?'), but this could vary among database vendors.</p>

.aut File	
[Settings] Field	Meaning for SQL Authentication
PasswordFormat	<p>By default, the PasswordFormat parameter is not listed in the [Settings] section of the .aut file.</p> <p>If no setting is given, Steel-Belted Radius expects the user's password in the SQL table to be in clear text.</p> <p>If set to 0, Steel-Belted Radius tries to determine password format automatically.</p> <p>If set to 3, Steel-Belted Radius expects the password value encrypted with UNIXcrypt.</p> <p>See "SQL Authentication and Password Format" on page 372.</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 database engine, which may or may not implement the feature.</p>
SQL	<p>Contains the SQL statement used to access the password information in the database. The SQL statement may 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. The subsequent lines may be indented for better readability.</p> <p>Example:</p> <pre>SQL=SELECT password, profile, fullname \ FROM usertable \ WHERE username = %name/63s</pre>
SuccessResult	<p>A string which is the expected result of a successful authentication, to be compared to the %result parameter.</p> <p>If this value is specified for this field, it is used in the following manner upon execution of the SQL statement: if the value of %result is not equal to the value given for this field, the user is rejected. The test for textual equality is not case sensitive.</p> <p>No such test, or rejection, is performed if no value is specified for this field.</p> <p>This is a useful technique for coordinating with the custom functionality of stored procedures.</p>
UpperCaseName	<p>Specifies whether the user's login name should be uppercased prior to using it in the SQL statement execution. Set this entry to 1 to convert the name to uppercase, set it to 0 to use the name exactly as received.</p>
WaitReconnect	<p>Specifies the number of seconds to wait after a failure of the database connection before trying to connect again.</p>

SQL Authentication [Strip] Sections

The [Strip] sections of the SQL authentication header file allow User-Name stripping to occur. That is, these sections enable Steel-Belted Radius to identify the username that the SQL database expects by stripping the incoming User-Name attribute value of realm names and other “decorations.”

You may or may not need to employ User-Name stripping for SQL authentication. Your need for this feature depends upon the naming conventions that you employ on your network and in your SQL database entries. Steel-Belted Radius’s usual name parsing features work independently of this feature.

See “Request Routing” on page 60.

The following [Strip] syntax is available to enable and configure User-Name stripping for SQL authentication:

```
[Strip]
Authentication=Yes
```

```
[StripPrefix]
String
String
```

```
.
.
.
```

```
[StripSuffix]
String
String
```

```
.
.
.
```

The meaning of these fields is as follows:

.aut File [Strip]	
Field	Meaning for SQL Authentication
Authentication	If set to Yes, prefix and suffix stripping is enabled for authentication packets. If No, it is disabled for authentication. If Authentication is set to Yes, when an authentication packet comes into the Steel-Belted Radius server and a SQL authentication method is active, stripping of the incoming User-Name attribute value occurs prior to SQL authentication as follows: First the prefixes listed in the [StripPrefix] section are stripped from the incoming User-Name attribute value, then the suffixes listed in [StripSuffix] are stripped. Then any other name processing that is appropriate at this point (for example, tunnel or proxy name parsing) is performed. Finally, the fully stripped name is authenticated against the SQL database.
[StripPrefix]	Lists strings that are to be stripped from the beginning of the User-Name value. The strings are listed in order of priority. A string that appears earlier in the list takes precedence. In the following example, if the incoming User-Name is "hitherebub", the stripped name is "bub". If the incoming User-Name is "hihowayya", the stripped name is "howayya": <pre>[StripPrefix] hithere hi</pre>
String	Each <i>String</i> that you provide in a [Strip] section may be a character string, or a regular expression according to the following rules: '?' is a wildcard character. A dash ('-') indicates a range of alphanumeric characters; brackets must enclose lists of characters or ranges. For example, [A-Za-z] means any letter and [0-9.] means any number, including decimal points and commas. A backslash ('\') followed by a non-alphanumeric character indicates that character literally, for example '\?' indicates the question mark. \ is also used as an escape character, as follows: <pre>\a bell (7) \b backspace (8) \t tab (9) \n newline (10) \v vertical tab (11) \f formfeed (12) \r return (13) \xnn hex value, where nn are 2 hex digits \nnn decimal value, where nnn are 3 decimal digits</pre>

.aut File [Strip]

Field	Meaning for SQL Authentication
[StripSuffix]	Lists strings that are to be stripped from the end of the User-Name value. Conventions are the same as for [StripPrefix].

Working with Stored Procedures in Oracle

You can write stored procedures for SQL that communicate with Steel-Belted Radius via input and output parameters to implement custom functions. Input parameters are the values passed to the procedure and output parameters are those set (or returned) by the procedure.

Exactly how you use these stored procedures, however, depends on details specific to the implementation of SQL that you are using. The following notes discuss some considerations specific to Oracle, which uses the term *package* and *package body* when referring to stored procedures.

Assume you have a `SELECT` statement that looks like the following:

```
SELECT password, profile, fullname FROM usertable WHERE
username = %name/63s
```

Let's say that you want to write a package called `myPack1` which performs the equivalent function. In this example, the package is defined as:

```
Package myPack1
    is
PROCEDURE myProc
(
    name IN VARCHAR2,
    pass OUT VARCHAR2,
    prof OUT VARCHAR2,
    fName OUT VARCHAR2
)
End myPack1;
```

The above is the Package and below is the Body. When referencing the package from `sqlauth.aut` you would point to the package name `myPack1` not `myProc`.

```
Package Body myPack1
    is
PROCEDURE myProc
(
    name IN VARCHAR2,
    pass OUT VARCHAR2,
```

```

        prof OUT VARCHAR2,
        fName OUT VARCHAR2
    )

IS

BEGIN

    SELECT password INTO pass, profile INTO prof, fullname
    INTO fName FROM usertable WHERE username = name;

END myProc;

End myPack1;

```

When you invoke the stored procedure, you should delineate each parameter as either an input (“!i”), output (“!o”), or input/output (“!io”) variable. Variables that are not specifically marked are considered by default to be input parameters.

You could replace the `SELECT` statement by invoking `myProc` as follows:

```

SQL=BEGIN myPack1.myProc(%name!i, %password!o,
    %profile!o, %fullname!o ); END;

```

When using input-output parameters with Oracle, you must set the `DefaultResults` setting to 0. Any other variables that need to be returned (such as `Reply-Message`) must be identified by the “!o” marker within the SQL statement.

See “SQL Authentication [Results] Section” on page 375.

SQL Accounting

12

- SQL Accounting
- Configuring SQL Accounting
- Connecting to the SQL Database
- SQL Statement Construction
- SQL Accounting Return Values
- SQL Accounting Header (.acc) File

SQL Accounting

Steel-Belted Radius can write RADIUS accounting information to an external SQL database, independently of the Steel-Belted Radius accounting log.

Note: SQL databases from several different vendors are supported.

To set up an external database for use as a repository for RADIUS accounting data, you must place an .acc database configuration file in the same directory that contains the Steel-Belted Radius service (normally C:\RADIUS\Service) or daemon. This file must be modified to contain specialized information about your enterprise database.

Steel-Belted Radius offers the SQL Accounting feature as a plug-in software module. Key features of the SQL plug-in include:

- The SQL statement is completely user-specified, allowing support of existing tables with existing field names and formats.
- It's SQL, so all kinds of arithmetic and string expressions may be part of the statement.
- The SQL statement is parameterized, so it is compiled once, and each execution uses variable data without need for recompilation.
- Attribute and other data from the accounting request may be easily mapped to any parameter of the SQL statement (and hence to any field in the table), using a simple syntax.
- Different request types may be mapped to different SQL statements that may operate against distinct tables within the database.
- Multiple executions of a SQL statement may be overlapped at the same time.
- Multiple instances of the SQL Accounting module may operate simultaneously, allowing logging to multiple databases.
- If the database connection drops, it is automatically reestablished after a configurable timeout, without the necessity of restarting Steel-Belted Radius.
- SQL Accounting responses can return information.
- Stored procedures invoked by SQL Accounting can make use of input parameters, record results, and return output parameters.

Important: While Steel-Belted Radius tries to provide uniformity in the operation of databases from different vendors, differences exist, particularly in the way SQL statements are interpreted. The capabilities of the SQL Authentication module depend on the capabilities of the underlying databases and their clients; things that work with one database may not work with another.

Configuring SQL Accounting

You must configure both Steel-Belted Radius and the SQL database to support SQL accounting. The exact configuration procedure must be tailored to the database that you use. However, all procedures must give the following results:

- The SQL server must be configured to be listening for client requests. Note that for SQL purposes, the Steel-Belted Radius server must be a client of the SQL server.
- The Steel-Belted Radius server must “know” about the remote SQL server. That is, it must know the machine where the SQL server software runs, and it must know the protocol and port used in communicating with that machine.
- The required transport must be in place between SQL client and server.

See “Configuring External Databases” on page 21.

Using the SQL Accounting Header File

To configure SQL Accounting, you must edit the accounting header file, `radsql.acc` (under **UNIX**) or `sqlacct.acc` (under **Windows**), located in the same directory that contains the Steel-Belted Radius service (normally `C:\RADIUS\Service`) or daemon. A reference listing of all header file options appears below. Most of these options may be left at their original settings; however, you must modify certain options to accommodate your own database.

See “Configuring External Databases” on page 21.

After you complete your `radsql.acc` (under **UNIX**) or `sqlacct.acc` (under **Windows**) changes and restart Steel-Belted Radius, accounting proceeds as you’ve configured it.

Using Multiple SQL Databases

You can configure Steel-Belted Radius to log accounting transactions against more than one SQL database.

To add an additional database, create a new header file with extension `.acc` in the same directory as `radsql.acc` (for **UNIX**) or `sqlacct.acc` (for **Windows**). You can give this file any name you like, provided its extension is `.acc`. At startup, Steel-Belted Radius enumerates all `.acc` files to create its list of accounting modules.

Important: *When creating the new file, start by copying the original `.acc` file, then make whatever modifications are necessary.*

Connecting to the SQL Database

Upon startup, the SQL Accounting module connects to the database, based on a connect string specified in the header file. The connect string contains information such as the name and location of the database, and the password required to connect. The connect string is passed to the database client to establish the connection.

While a sample connect string is provided in the original header file, you must configure the Connect entry of the header file with a connect string appropriate to your database.

The password for database access must be provided as part of the connect string or the following results occur:

- Under **UNIX**, the connection fails.
- Under **Windows**, at startup and each time a reconnect is required, a pop-up dialog prompts the user to enter a password before making the connection.

If the initial attempt to connect to the database fails, or if in the course of processing an error occurs that the SQL Accounting module interprets as a database connection failure, the SQL Accounting module drops the connection and attempts to establish a new connection after a period of time. In the interim, all authentication requests are ignored.

The SQL Accounting module uses an exponential back-off strategy in determining how long to wait before attempting a new connection, as well as how frequently this attempt should be made. After the first dropped connection, it waits a certain amount of time before attempting to reconnect. If this attempt to reconnect also fails, it waits for twice the amount of time before trying again; and so on, up to some maximum wait time. The initial and maximum wait times are configurable.

*Warning: (UNIX only): Detailed error information may not be available if there is an error processing the database logon at connect time. A numeric result code appears in the log. You may need to refer to product-specific documentation to decode this result code. With Oracle on UNIX, you can use the **oerr facility-code error-number** command with a facility code of **ora** from the UNIX command shell.*

SQL Statement Construction

For each accounting request whose Acct-Status-Type is mapped to a SQL statement, that accounting request is logged to the backend database by executing the associated SQL statement.

While a sample SQL statement is provided in the original header file, you must configure one or more SQL entries of the header file with a statement appropriate to your database. Each SQL statement is typically an `INSERT INTO` statement and may contain additional syntax elements that are preprocessed by the SQL Accounting module.

The SQL Accounting module executes SQL statements in parameterized form. This means that the SQL statement is compiled once, with parameter markers (usually question marks) as placeholders for data items that vary from one execution to the next. Only upon execution of the statement are the actual data values supplied.

The SQL statement you compose must not include parameter markers directly. Instead, the names of the parameters should be included where parameter markers would appear, in a format described below. The SQL Authentication module translates the SQL statement provided, replacing parameter names with parameter markers prior to passing the SQL statement to the database engine.

A SQL statement can be very simple. Basically, all that is required is to set fields of the database record with values from the request. The SQL statement can also be quite complex; it can include inner joins, and it can contain expressions. The underlying database engine is responsible for handling the SQL statement; The SQL Accounting module performs no interpretation of the SQL statement other than to translate parameter names to parameter markers.

INSERT Statement and VALUES Section

The following is an example of a SQL `INSERT` statement that might be found in a Steel-Belted Radius `.acc` file:

```
INSERT INTO usagelog (Time, NASAddress, SessionID, Type,
Name, BytesIn, BytesOut) VALUES (%TransactionTime,
%NASAddress, @Acct-Session-Id, @Acct-Status-Type,
%FullName/40s, @Acct-Input-Octets, @Acct-Output-Octets)
```

In the `VALUES` section, the names (between parentheses) represent the values inserted into the SQL table columns. To support the SQL Accounting module, each item in the `VALUES` section must be prefixed with either an at sign (`@`) or a percent sign (`%`):

- `@` indicates a RADIUS accounting attribute. The attribute name must also be listed in the `account.ini` file. This remains true even if the `account.ini` file is disabled.

- ‘%’ indicates an item associated with the `INSERT` request that is not a RADIUS accounting attribute. The following Steel-Belted Radius specific items may be provided:

Item	Data Type	Meaning
%TransactionTime	Time	The date/time that the event occurred that is the subject of the request.
%Time	Time	The date/time when the request is being processed. (This is later than %TransactionTime if the request is a retry.)
%Type	String	The RADIUS accounting request type. For important details, see “SQL Accounting [TypeNames] Section” on page 399.
%NASAddress	IP address	The IP address of the requesting NAS.
%NASName	String	The name of the requesting NAS.
%NASModel	String	The NAS make/model.
%FullName	String	The full name of the logged in user.
%AuthType	String	The method by which the user was authenticated.

A format specifier may appear immediately following each parameter. The format specifier should describe the database storage format of the column that corresponds to the parameter. It consists of a slash (‘/’), possibly a length, and a data type. The following data types are available:

Format Specifier	Meaning
/xs	A text string of length x. /s indicates a string with the default length of 256
/xb	A binary data string of length x. A binary string is different from a text string in that it is not NULL-terminated and is not restricted to ASCII characters. /b indicates a binary data string with the default length of 256.
/n	32-bit integer
/n8	8-bit integer
/n16	16-bit integer
/n32	32-bit integer (same as /n)
/t	Timestamp

Note: Steel-Belted Radius supports integers larger than 32 bits by manipulating them as binary data strings. The Solaris Oracle 8 plug-ins are able to correctly convert binary data strings between Oracle `VARRAW` types (`/xb`) and Oracle `NUMBER` types (`/n`). Oracle types must be declared with enough precision to avoid truncation when inserting into the database, and care

must also be taken to avoid truncation when retrieving from the database. In particular, avoid retrieving Oracle VARRAW types larger than 256 bytes. Other database/ operating-system combinations may not allow for integers larger than 32 bits.

If a format specifier is not present in the SQL statement syntax, Steel-Belted Radius automatically defaults to an appropriate specifier based on the actual parameter type. For example, @Acct-Input-Octets is a number, and defaults to /n.

Warning: For strings, always include a format specifier, and be sure to specify a length no greater than the actual field size in the database. The compilation of the SQL statement may fail if a length greater than the actual field size is specified. If no format specifier is present, the length defaults to 256 characters, which may cause the compilation to fail.

Steel-Belted Radius automatically attempts to convert between the internal format of a parameter and its format in the database, as described by the format specifier. In most cases, the formats are equivalent; if not, Steel-Belted Radius performs reasonable conversions.

The following table lists the internal formats and their compatible database formats:

Internal Format	Compatible Database Formats
Binary data string	/b, /xb, /n, /n8, /n16, /n32
Number	/n, /n8, /n16, /n32, /xs, /s
String	/xs, /s
Time (seconds since 1/1/70)	/t, /n, /n32, /xs, /s
IP address	/n, /n32, /xs, /s

As you write the INSERT statement for your SQL accounting header file (.acc), we recommend the following syntax checklist:

- The column names and their corresponding attributes in the VALUES section are order-dependent. In the example above, the %TransactionTime value would be inserted into the Time column, the %NASAddress value would be inserted into the NASAddress column, and so forth. The ordering of these settings is critical to proper RADIUS accounting data insertion, since each column in the SQL table may be a unique data type (varchar, int, and so forth).
- The use of left and right parentheses ‘()’, the backslash ‘\’, the forward slash ‘/’ and even blank spaces are all extremely important and must be exact. You can add as many columns and attributes as you want for your RADIUS accounting needs; however, be sure to model your INSERT statement syntax on the example above.

- An attribute listed in the `VALUES` section incorrectly, for example `@Acct_Session-Id` rather than `@Acct-Session-Id`, causes the SQL statement to fail during a RADIUS accounting transaction. The attribute's syntax must match its corresponding attribute name in the `account.ini` file, which in turn matches the attribute's name in the appropriate dictionary file, which allows Steel-Belted Radius to process the attribute correctly when it is received from the NAS (the RADIUS client).
- An attribute listed in the `VALUES` section that is missing its prefix of '@' or '%' causes the SQL statement to fail during a RADIUS accounting transaction.
- If a carriage return is present within the `INSERT` statement without the backslash '\ ' to indicate the end of the line, the SQL statement fails during a RADIUS accounting transaction.
- Don't make the lines in the `.acc` file too long. There is a line length limit of 255 characters. Use the backslash '\ ' to indicate the end of the line before that limit is reached. If a line exceeds this limit, the SQL statement fails during a RADIUS accounting transaction.

Using Multiple SQL Statements

The most common use of accounting is to track user sessions. However, there are also accounting requests generated when the NAS starts up and shuts down; and, there are vendor-specific uses of accounting that track other NAS phenomena as well. Clearly, it might be advisable to log different types of accounting events to different tables.

The `Acct-Status-Type` attribute of an accounting request indicates the request type. You may, if you like, create multiple SQL statements, and map each `Acct-Status-Type` to one of these SQL statements. The different statements may update different tables in the database, but they all share the single database connection.

Overlapped Execution of SQL Statements

The SQL Accounting module is multi-threaded. SQL Accounting can be configured with a maximum number of simultaneous executions of any SQL statement, using the `MaxConcurrent` entry in the `.acc` file's `[Settings]` section.

If `MaxConcurrent` is set to 1, SQL execution occurs serially, and the SQL execution for each accounting request must complete before execution for the next request may begin.

By increasing `MaxConcurrent`, it may be possible to increase throughput by overlapping operations, especially if the database server is remote and a large part of the time to complete a statement execution is taken up by network latency. If the database server is local, the point of diminishing returns may be reached at a small value of `MaxConcurrent`, possibly even at 1 or 2. You can find the optimum value for your system by experimentation.

***Important:** A setting of `MaxConcurrent = 1` should be sufficient for all but the most demanding environments. Increase this value only slowly and conservatively.*

`MaxConcurrent` determines the maximum overlap for executing any single SQL statement. Multiple SQL statements for different request types are not interdependent, and executions of one statement do not affect executions of a different statement.

You might expect that databases that are licensed by number of connections would debit a single connection regardless of how many SQL statements are active. This is not necessarily the case; some databases count each open compiled SQL statement against the licensed number of connections. The database license may also have an influence on the optimum setting for `MaxConcurrent`.

SQL Accounting Return Values

SQL Accounting statements can now return information (i.e., RADIUS attributes) in an accounting response. This is useful only if you are using a client that expects and supports attributes embedded in a RADIUS accounting response message.

Stored procedures can also return output parameters. The way in which these stored procedures are called depends on your operating system:

- To call an Oracle stored procedure in a **UNIX** environment:

```
BEGIN storedProcedure(parameters...); END;
```

- Under **Windows**:

```
call (storedProcedure(parameters...))
```

SQL Accounting Header (.acc) File

The header file used to configure the SQL Accounting module must have the extension `.acc`. The format of a header file is comparable to that of a Windows INI

file. It is composed of several sections; each section may contain multiple entries. Section names are enclosed in square brackets; entries are of the form *attribute=value*.

SQL Accounting [Bootstrap] Section

The [Bootstrap] section of the SQL accounting header file specifies information used to load and start the SQL Accounting module.

.acc File	
[Bootstrap] Field	Meaning for SQL Accounting
LibraryName	This entry must be set to the name of the SQL accounting module. Under UNIX , this is <code>radsql_auth_ora.so</code> (for Oracle) Under Windows , this is <code>SQLACCT.DLL</code> .
Enable	This entry must contain a 1 to enable the module, 0 to disable it. Upon installation, this entry is set to 0; to enable SQL Accounting, you must change this entry to 1.
InitializationString	This entry is unused.

SQL Accounting [Settings] Section

The [Settings] section of the SQL accounting header file defines parameters that control the database connection.

.acc File	
[Settings] Field	Meaning for SQL Accounting
ConcurrentTimeout	Specifies the number of seconds a request may wait for execution before it is discarded. Since there may be up to <code>MaxConcurrent</code> SQL statements executing at one time, as new requests arise they must be queued, waiting for other statements to complete. ConcurrentTimeout may be overridden for any particular statement in the <code>[Type/statement]</code> section for that statement.
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. The exact format of the connect string depends on the database you use: see the configuration instruction file in the same directory that contains the Steel-Belted Radius service or daemon.

.acc File	
[Settings] Field	Meaning for SQL Accounting
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 may or may not implement the feature.
MaxConcurrent	Specifies the maximum number of instances of a single SQL statement that may be executing at one time. MaxConcurrent may be overridden for any particular statement in the [Type/ <i>statement</i>] section for that statement.
MaxWaitReconnect	Specifies the maximum number of seconds to wait after successive failures to reconnect after a failure of the database connection. WaitReconnect specifies the time to wait after failure of the database connection. This value is doubled on each failed attempt to reconnect, up to a maximum of MaxWaitReconnect.
ParameterMarker	The character or sequence of characters used as the parameter marker in a parameterized SQL query. Normally, this is the question mark ('?'), but this could vary among database vendors. The parameter marker is ignored by the Solaris Oracle authentication method which uses ':1', ':2', and so forth.
QueryTimeout	Specifies the number of seconds to wait for the execution of a SQL statement to complete before timing out. This value is passed to the database engine, which may or may not implement the feature. QueryTimeout may be overridden for any particular statement in the [Type/ <i>statement</i>] section for that statement.
UpperCaseName	Specifies whether the user's login name should be uppercased prior to using it in the SQL statement execution. Set this entry to 1 to convert the name to uppercase, set it to 0 to use the name exactly as received.
UTC	This entry should be set to 0 to show time information in local time, or 1 to show time information in universal time coordinates (UTC).
WaitReconnect	Specifies the number of seconds to wait after a failure of the database connection before trying to connect again.

SQL Accounting [Type] Sections

Each entry in the [Type] section of the SQL accounting header file maps an Acct-Status-Type attribute value to a statement name that you may assign arbitrarily. The statement name is then used to look up another section in the header file that describes that statement. The secondary section names are composed as follows:

[Type/*statement*], where *statement* is the arbitrarily assigned name for the statement.

For example, to perform separate accounting updates for NAS and user activity, you might provide the following [Type] and [Type/*statement*] sections:

```
[Type]
1=user
2=user
3=user
7=nas
8=nas
639=nas
28=nas

[Type/user]
SQL=INSERT INTO usagelog \
    (Time, NASAddress, SessionID, \
    Type, Name, BytesIn, BytesOut) \
VALUES \
    (%TransactionTime, %NASAddress, \
    @Acct-Session-Id, @Acct-Status-Type, \
    %FullName/40s, @Acct-Input-Octets, \
    @Acct-Output-Octets)

[Type/nas]
SQL=INSERT INTO . . .
```

Note the numeric values used in the [Type] section above. The Acct-Status-Type values 1, 2, 3, 7, and 8 have been reserved by the RADIUS accounting standard with names and meanings as follows:

Acct-Status-Type Value	Name	Meaning
1	Start	A user session has started
2	Stop	A user session has stopped, request contains final statistics
3	Interim	A user session is in progress, request contains current statistics
7	Accounting-On	The NAS has started up
8	Accounting-Off	The NAS is about to shut down

Additional values for Acct-Status-Type have been defined by NAS vendors for use with their equipment. These vendor-specific values may also be listed in the [Type] section.

SQL Accounting [Type/statement] Sections

The following fields may be present in a [Type/statement] section of the SQL accounting header file:

.acc File	
[Type/statement] Field	Meaning for SQL Accounting
SQL	<p>The SQL field provides the exact SQL statement used to update the SQL database with accounting information. The SQL statement may 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. The subsequent lines may be indented for better readability. For example:</p> <pre>SQL=INSERT INTO usagelog \ (Time, NASAddress, SessionID, \ Type, Name, BytesIn, BytesOut) \ VALUES \ (%TransactionTime, %NASAddress, \ @Acct-Session-Id, \ @Acct-Status-Type, \ %FullName/40s, \ @Acct-Input-Octets, \ @Acct-Output-Octets)</pre> <p>See also “SQL Statement Construction” on page 368.</p>
MaxConcurrent	<p>If present, MaxConcurrent overrides the value of MaxConcurrent specified in the [Settings] section for this particular statement.</p>
ConcurrentTimeout	<p>If present, ConcurrentTimeout overrides the value of ConcurrentTimeout specified in the [Settings] section for this particular statement.</p>
QueryTimeout	<p>If present, QueryTimeout overrides the value of QueryTimeout specified in the [Settings] section for this particular statement.</p>

SQL Accounting [TypeNames] Section

Each entry in the [TypeNames] section of the SQL accounting header file maps an Acct-Status-Type attribute value to a string. If a %Type parameter is present in the corresponding SQL statement, this %Type parameter contains the given string.

If no string is given for a particular Acct-Status-Type, when an accounting request of that type is received, %Type is set to the numeric value of the Acct-Status-Type attribute, formatted as a string.

The syntax for the [TypeNames] section is as follows:

```
[TypeNames]
TypeID=TypeName
TypeID=TypeName
.
.
.
```

You can include RADIUS standard and vendor-specific accounting packet types; for example:

```
[TypeNames]
1=Start
2=Stop
3=Interim
7=On
8=Off
639=AscendType
28=3ComType
```

Working With Stored Procedures

The SQL example in the previous section could be replaced by a custom stored procedure. This stored procedure might look something like the following:

```
PROCEDURE myProc
(
    ttime    in    varchar2,
    nasaddr  in    varchar2,
    sessid   in    varchar2,
    ttype    in    varchar2,
    uname    in    varchar2,
    bytein   in    varchar2,
    byteout  in    varchar2
);
END myProc;

CREATE OR REPLACE PACKAGE BODY myPack1 IS
    PROCEDURE myProc
    (
        ttime    in    varchar2,
        nasaddr  in    varchar2,
        sessid   in    varchar2,
        ttype    in    varchar2,
        uname    in    varchar2,
        bytein   in    varchar2,
        byteout  in    varchar2
    )
```

```

)
IS
BEGIN
    INSERT INTO usagelog
        ( Time, NASAddress, SessionID, Type, Name,
          BytesIn, BytesOut )
    VALUES
        ( ttime, nasaddr, sessid, ttype, uname, bytein,
          byteout );
    END myProc;
END myPack1;

```

When you invoke the stored procedure, you should delineate each parameter as either an input (“!i”), output (“!o”), or input/output (“!io”) variable.

This stored procedure can be invoked with the following connect string in the radsq1.acc file:

```

SQL=BEGIN myPack1.myProc(%TransactionTime!i,
    %NASAddress!i, @Acct-Session-Id!i, %Type!i,
    %FullName!i, @Acct-Input-Packets!i,
    @Acct-Output-Packets!i); END;

```

Load Balancing Example

The following excerpt from an .acc example file configures load balancing between two SQL servers (so that the work load is shared nearly equally between two servers). The tradeoff with this technique is that the data is split between two servers and must be reintegrated when processed. For example, the Accounting-START for an end-user may be stored on one server and the corresponding Accounting-STOP on the other.

```

[Server]
s1=2
s2=2

[Server/s1]
Connect=system/*****@thor

[Server/s2]
Connect=system/*****@odin

[Type]
1=User
2=User
3=User

[Type/User]

```

```
SQL=INSERT INTO acct1(TransTime, FullName, \  
    Authenticator, NASName, NASAddress, Type, \  
    PacketsIn, PacketsOut) \  
VALUES (%TransactionTime, %FullName/40s, \  
    %AuthType/40s, %NASName/40s, %NASAddress, \  
    %Type, @Acct-Input-Packets/n, \  
    @Acct-Output-Packets/n)
```

LDAP Authentication

13

- External LDAP Authentication
- LDAP Authentication Header (.aut) File
- LDAP Authentication Sequence
- LDAP Authentication Examples

External LDAP Authentication

The Steel-Belted Radius server can authenticate against records stored in an external LDAP database. Any attribute(s), such as username and password, can be used to query the database.

External database authentication is normally used when an organization already has a large amount of user information stored in an LDAP database, and wants to authenticate these users using RADIUS. Authentication against an existing LDAP database extends authentication services to user accounts without requiring an administrator to enter user information into the Steel-Belted Radius database.

Steel-Belted Radius offers the LDAP Authentication feature as a plug-in software module. Key features of the LDAP plug-in include the following:

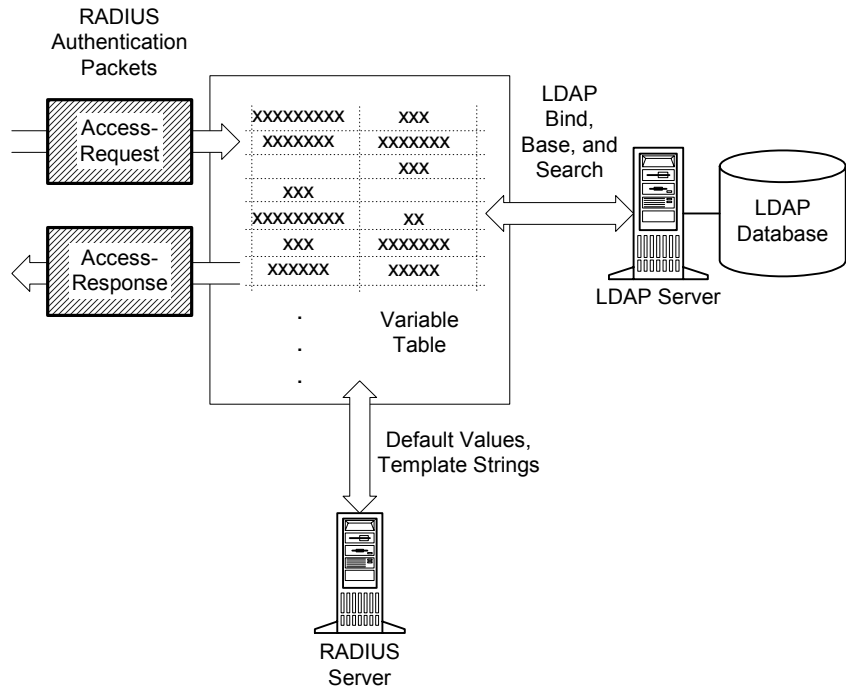
- LDAP Version 3 is supported.
- SSL is supported if you want to use it (requires Netscape certificates).
- You can authenticate via LDAP Bind or via a password returned from an LDAP Search request (BindName).
- A single Search request or a sequence of Search requests may be specified.
- Bind, Base and Search strings may include variables.
- New Bind parameters can be specified during a sequence of searches.
- Other authentication credentials can be specified in a string that can include variables.
- Variables may be set from the RADIUS request packet and from LDAP Search results.
- Variables may be used to specify RADIUS response attributes and other response information.
- The RADIUS response can include RADIUS attributes found in the LDAP database, or it can reference a Steel-Belted Radius profile or user entry.
- Several features similar to SQL authentication are supported, such as round-robin load balancing, the “server of last resort,” activation targets, and so on.
- Decorated usernames (`attribute 01 User-Name`) can be parsed into two variables within the variable table. For example, `simon@xyz.com` would be parsed into `simon` and `xyz.com` for use later in the authentication process.
- The variable table allows both attributes and `%Profile` in the [Response] section.

- Branching is supported using the `OnFound` and `OnNotFound` fields. This feature provides powerful data lookup and authorization options.

LDAP Variable Table

The central mechanism that allows you to connect RADIUS information with LDAP information is the Variable Table.

At the beginning of each LDAP authentication request, Steel-Belted Radius creates a Variable Table. Attributes and other information from the RADIUS request are entered in the Variable Table for use in LDAP Bind, Base, and Search strings. When attributes are returned by LDAP requests, they too are entered in the Variable Table. Finally, selected information from the Variable Table is returned to the RADIUS client in the RADIUS response packet.



Role of the Variable Table in LDAP Authentication

Types of LDAP Authentication

To design an LDAP authentication method, consider how you want to validate the username and password.

The LDAP plug-in offers two techniques for validating the username and password. Each header file that you write to control LDAP authentication must employ either

one technique or the other: Bind, or BindName. The differences between the two techniques have to do with (1) how Steel-Belted Radius connects to the LDAP server and (2) whether the username/password validation is performed by the LDAP server or by Steel-Belted Radius.

BindName Authentication

In the BindName case, your LDAP header file provides Steel-Belted Radius with the username and password of an account on the LDAP server. This must be an account that has privileges to access all of the information that you require to authenticate users. In the LDAP header file, you provide the username in the BindName parameter, and the password in the BindPassword parameter.

After you complete the LDAP header file, each time the Steel-Belted Radius server starts up, it executes a Bind request to the LDAP server using the BindName and BindPassword parameters as its credentials. If the LDAP server can validate these credentials, a connection is established between the two servers. This connection remains “up” all the time. It is disconnected only if the Steel-Belted Radius server or the LDAP server goes down, and it’s re-established as soon as possible after the “down” server comes back up. The LDAP header file offers a number of connection and re-connection timeouts and other parameters that regulate this relationship.

Any time authentication via LDAP is required, Steel-Belted Radius consults the corresponding LDAP header file. In the BindName case, this file must contain a Search command that maps the username from the Access-Request to a password attribute in the LDAP database. The Search may retrieve other LDAP attributes as well. When the Search returns its results, Steel-Belted Radius compares the value of the password returned from the LDAP database with the password from the incoming Access-Request. If the two values are the same, the password is considered validated.

When the connection to the LDAP server is established using BindName, multiple authentications can be performed at the same time over the same connection.

Bind Authentication

In the Bind case, Steel-Belted Radius authenticates connection requests by attempting to Bind to the LDAP server using the username and password from the incoming Access-Request or from a configured username and password. If this Bind request succeeds, the password is validated. This is essentially “pass-through” authentication; Steel-Belted Radius presents an LDAP user’s credentials to the LDAP server and asks to have them validated.

In the simplest case, a single connection is established for each Access-Request and is kept open only long enough for the LDAP server to validate the password and

respond to any Search requests. Then Steel-Belted Radius closes the connection and completes any processing that remains to generate an Access-Response.

A more sophisticated search technique can take advantage of flexible Bind, which allows you to allocate a sequence of connections for each Access-Request. Each in turn is kept open only long enough for the server to process each search criterion. Then Steel-Belted Radius closes the connection and completes any processing that remains to generate an Access-Response.

Attributes and LDAP Authentication

A username and password may be all the information that you require to authenticate users. However, the LDAP plug-in offers a number of techniques for working with Check-List and/or Return-List attributes, should you need them.

See “User Attribute Lists” on page 54.

See also “LDAP Authentication Header (.aut) File” on page 412.

Configuring LDAP Authentication

To configure an LDAP authentication method, you must edit the header file that controls the LDAP authentication sequence.

See “LDAP Authentication Header (.aut) File” on page 412.

The bulk of this chapter explains the meaning and syntax of each section in the LDAP authentication header file. The order of topics is as shown in the following table. This table traces the process of configuring an LDAP authentication method for Steel-Belted Radius, step by step. It also lists the sections that you must edit in the header file to accomplish each step. No step may be omitted. You must at least consider the entries that you want to put in each section of the header file, even if you end up deciding to leave most of that section blank.

Step	LDAP Configuration Task	.aut File Sections
1	Decide how you want Steel-Belted Radius to validate RADIUS access requests. There are two major areas of choice, as described above: (1) Bind or BindName; and (2) Profile, Alias, or attribute list.	All sections
2	Determine which incoming RADIUS attributes support your desired response.	[Response]
3	Determine which LDAP attributes support your desired response.	[Attribute/name]
4	Design Search template(s) that can find the necessary data in your LDAP database schema.	[Search/name]

Step	LDAP Configuration Task	.aut File Sections
5	Extract the data from the incoming RADIUS packet that Steel-Belted Radius must support the LDAP Bind and Search requests.	[Request]
6	Determine defaults that you want Steel-Belted Radius to use when corresponding values are not provided.	[Defaults]
7	Enable connections between the Steel-Belted Radius server and LDAP server(s).	[Server] [Server/name] [Settings] [Failure]
8	Enable the LDAP plug-in and name the authentication method.	[Bootstrap]

The order in which you should edit header file sections is almost exactly the reverse of the order in which Steel-Belted Radius processes them. The processing sequence is described in “LDAP Authentication Sequence” on page 430. Feel free to take a look, but it will make more sense after you've examined the header file in detail.

Supporting Secure Sockets Layer

You must follow the instructions below for SSL to be supported by the LDAP plugin:

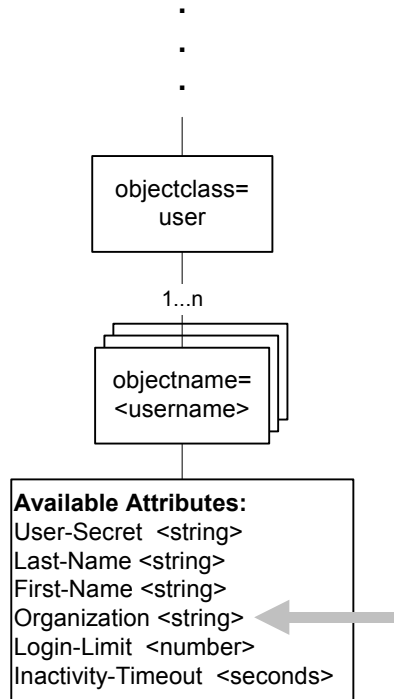
- 1 Set `SSL` in the [Settings] (or [Server/name]) section to 1.
- 2 Set the `Certificates` field in the [Settings] (or [Server/name]) section to the path where the `cert7.db` file is located. The name of the file (e.g., `cert7.db`) should not be included.
- 3 Set the port in the [Server] section to the SSL port of the LDAP server.

LDAP Database Schema

The most important factor in the success of your LDAP authentication methods is the design of your LDAP database schema. It's assumed that you already have a schema in place.

Often, you can use the LDAP plug-in *without* changing the LDAP database schema at all. In the figure below, the user record already provides an LDAP attribute called `Organization`. If you intend to grant connection privileges according to which organization each user belongs to, you can create profiles in the Steel-Belted Radius database whose names match the strings you are already using for the `Organization` attribute. Then you can create an LDAP authentication header file that retrieves the value of the `Organization` attribute from the LDAP database and returns it to Steel-Belted Radius as the name of the profile to use.

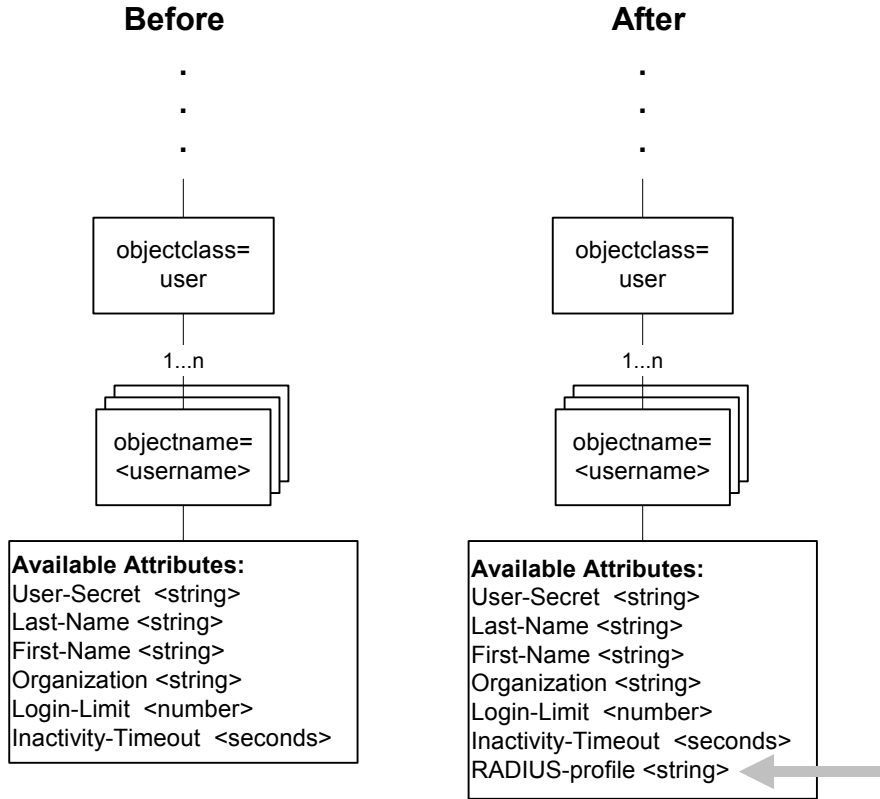
Note: If you are using BindName authentication, you need to be able to identify which LDAP attribute contains the user's password. In the schema below, this attribute is called User-Secret.



Capitalizing on an Existing Schema for LDAP Authentication

In some situations, you might want to modify the schema, for example if the authentication strategy you've chosen requires data that is not currently in the schema.

The name of a Steel-Belted Radius profile is a typical example. Consider the example above. If you want to assign connection privileges to users in some way other than by Organization, and there is no other LDAP attribute that seems appropriate, you can add an LDAP attribute that names a profile. In the figure below, this attribute is called RADIUS-Profile. This attribute contains a string value that can be set to the name of a profile defined in the Steel-Belted Radius database.



Modifying a Schema to Enhance LDAP Authentication

This said, LDAP concepts and the details of your own LDAP schema are entirely outside the scope of this chapter. The instructions in this chapter are provided to help you to make the LDAP plug-in work with an existing LDAP database or databases to provide a Steel-Belted Radius authentication method. The instructions assume that you already have a working knowledge of LDAP syntax and conventions.

For details about LDAP, please refer to your usual LDAP information source.

LDAP Authentication and Password Format

Steel-Belted Radius supports authentication of users whose records reside in an LDAP, Native, and SQL table, in which password values are stored in one of the following formats: clear text, UNIXcrypt, Secured Hash Algorithm (SHA1+Base64 hash), MD4 hash, or enc-md5 reversibly-encoded password.

Hashed Passwords

Encoded values include a prefix that indicates how the password has been processed. The prefix is in clear text between curly braces ‘{’ ‘}’ and is immediately followed by a hash value computed from the password. If no prefix is present in the value retrieved, the entire password is assumed to be in clear text format. In summary:

- `PasswordText` indicates clear text format (no encryption)
- `{crypt}HashHash` indicates UNIXcrypt format
- `{SHA}HashHashHash` indicates SHA1+Base64 hash encryption
- `{md4}HashHash` indicates MD4 hash of the Unicode form of password

Note: Refer to RFC 2759 for details about how MS-CHAP-V2 produces an MD4 hash value.

- `{enc-md5}EncryptedEncrypted` indicates a reversibly encrypted password

Note: Although Steel-Belted Radius reads passwords encoded in this format, you must purchase the Software Developer’s Kit to convert clear-text passwords to this format.

UNIXcrypt is the standard hash algorithm that is used for the `/etc/passwd` file on UNIX systems. This may be necessary if, for example, the standard user database on a UNIX machine (the `/etc/passwd` file) is migrated to a SQL database, so that the values in the `Password` column of the SQL table are processed with UNIXcrypt.

Steel-Belted Radius may be configured to expect that the values retrieved from a table have been run through UNIXcrypt by adding the following entry into the [Settings] section of the LDAP authentication header file:

```
PasswordFormat=3
```

Automatic Parsing

If `PasswordFormat` is set to 0, Steel-Belted Radius attempts to determine the password format automatically by parsing it. This is the recommended setting. Automatic parsing expects the password to be stored in one of the formats above.

This technique is useful if clear text passwords are available to the Steel-Belted Radius server (that is, if PAP is used). If you set `PasswordFormat` to 0, the stored password can be returned to Steel-Belted Radius still encrypted, and the comparison with the password received from the RADIUS client can be done on the Steel-Belted Radius side.

Note: The setting for automatic password parsing in previous versions of Steel-Belted Radius (`auto`) has been deprecated.

LDAP Authentication Header (.aut) File

The LDAP authentication header file is located in the same directory that contains the Steel-Belted Radius service (normally C:\RADIUS\Service) or daemon. The header file must have the extension .aut and is usually called ldapauth.aut.

The format of the LDAP authentication header file is comparable to that of a Windows INI file. It is composed of several sections; each section may contain multiple entries. Section names are enclosed in square brackets, for example [Bootstrap]. Each entry in the section appears all on one line, and is of the form *field = value*. A section ends at the next section, or at the end of the file. Everything to the right of a semicolon (;) is ignored until the end of that line.

LDAP Authentication Variable Names

When Steel-Belted Radius extracts RADIUS attribute values from the incoming Access-Request and adds them to the Variable Table, the name that it gives to each variable is the same as the name of the corresponding attribute, for example User-Name or Calling-Station-ID. You may refer to the variable by this name in any subsequent entry in the .aut header file. This convention means that RADIUS attribute names are treated as reserved keywords. However, the .aut header file syntax also permits you to assign the value of an incoming RADIUS attribute to any variable.

When the LDAP Search request returns LDAP attribute values, they too are added to the Variable Table. Steel-Belted Radius gives each variable the same name as the corresponding LDAP attribute. In the schema illustrated above, this would produce variable names such as User-Secret and Last-Name. For the correct names to use in your own .aut header file, you'll need to consult your LDAP database schema. Like RADIUS attribute names, LDAP attribute names are treated as reserved keywords. However, the .aut header file syntax also permits you to assign the value of a returned LDAP attribute to any variable.

LDAP Authentication [Response] Section

During an authentication transaction, the [Response] section is the last section in the LDAP authentication header file to be processed. At this point in processing, all Bind and Search requests to the LDAP database have been completed.

The [Response] section tells Steel-Belted Radius what to do with the information that it has retrieved from the incoming Access-Request and from the LDAP database. The goal at this point is for Steel-Belted Radius to complete authentication and issue an Access-Response to the RADIUS client.

The [Response] section syntax is as follows:

```
[Response]
  attribute = variable
  attribute = variable
  .
  .
  .
```

where *attribute* is the name of a RADIUS attribute or other special item needed to complete authentication, and *variable* is the name of a variable in the Variable Table. The end result of the [Response] syntax is that the value in the variable is assigned to the attribute.

An IP pool can be returned for any attribute of the appropriate type. If the returned string appears to be an address (i.e., in the format, a.b.c.d), it is considered an address; otherwise, it is considered a pool, from which an address is allocated.

attribute may be the name of a RADIUS attribute, or it may be one of the following keywords, which identify various special items associated with Steel-Belted Radius. Each of these keywords begins with the percent sign (%) to distinguish it clearly from the RADIUS attributes.

Item	Meaning for LDAP Authentication
%LoginLimit	The name of the variable specifying the Maximum Concurrent Connection limits.
%Password	For BindName authentication, you must provide a %Password entry in the [Response] section and you must assign it the value of the password attribute retrieved from the LDAP database. Steel-Belted Radius validates the password received in the Access-Request by comparing it with the value assigned to %Password. If the passwords don't match, the request is rejected. <i>NOTE: The user's password may be in clear text, or encrypted with UNIXcrypt or a SHA1+Base64 hash.</i> See "LDAP Authentication and Password Format" on page 410. For Bind authentication, omit %Password. Once processing reaches the [Response] section, the password has already been validated.
%Profile	The name of a Profile entry in the Steel-Belted Radius database. If the password has been validated (by BindName or Bind), with %Profile listed in the [Response] section, then %Profile may be set to any variable, for example: %Profile = userpolicy

Item	Meaning for LDAP Authentication
%ProxyRealm	<p>When the search filter is set to find a user or object in the LDAP database that includes the <code>userpolicy</code> LDAP attribute, this value is retrieved and returned to the Steel-Belted Radius database so that it may be matched with an existing Profile entry of the same name.</p> <p>If the value of <code>userpolicy</code> is "prof1" and a Profile called <code>prof1</code> exists in the Steel-Belted Radius database, any Return-List or Check-List attributes in <code>prof1</code> are applied to the user's connection.</p> <p>If the value returned from LDAP cannot be matched with an existing Profile in the Steel-Belted Radius database, the user is rejected due to "Insufficient Resources."</p> <p>The realm to which the authentication must be proxied. If ProxyRealm is not set, Routed Proxy does not occur. See "Routed Proxy" on page 461.</p>
%ProxyUserName	<p>The User-Name attribute, which must be sent in the proxy request. If ProxyUserName is not set, the User-Name from the original request packet is used. See "Routed Proxy" on page 461.</p>
%Alias	<p>The name of a Native User entry in the Steel-Belted Radius database.</p> <p>If the password has been validated (by BindName or Bind), with %Alias listed in the [Response] section, then %Alias may be set to any variable, for example:</p> <p style="padding-left: 40px;">%Alias = userpolicy</p> <p>Important: You are strongly recommended to use %Profile, as use of %Alias has been deprecated. The %LoginLimit value allows you to implement the concurrent connection limits previously available through %Alias.</p> <p><i>Note: Native User entries without passwords automatically cannot be authenticated. This is a safety feature built into Steel-Belted Radius. Therefore, setting up Native User entries in preparation for using the Alias parameter with LDAP authentication does not pose a "back door" security risk.</i></p> <p>Generally, even if a very large number of users reside in the LDAP database, you need to add only one or two Native User entries to the Steel-Belted Radius database. The concurrent connection limit associated with a single Native User entry may be applied to any number of users in the LDAP database. Often a Native User entry with a connection limit of 1, and a second Native User entry with a connection limit of 2, is sufficient for the entire LDAP database.</p> <p>For example, analog users may be allowed a connection limit of 1, while ISDN users are allowed a connection limit of 2.</p>

Item	Meaning for LDAP Authentication
	<i>NOTE: The Native User authentication method displayed in the Admin Configuration dialog does not need to be activated for the Alias feature to work.</i>
%FullName	The fully distinguished name of the User, for Steel-Belted Radius accounting purposes. This is the exact name against which authentication was performed. Depending on what may have occurred during Steel-Belted Radius name parsing, this name may or may not be different from the value of the User-Name attribute as it originally arrived in the Access-Request.

LDAP Authentication [Attributes/*name*] Sections

LDAP database entries may have many attributes, many of which may be irrelevant to the authentication process. An LDAP Search returns all of the attributes associated with an LDAP entry. Therefore, when specifying an LDAP Search for authentication purposes, you may want to provide a list of specific LDAP attributes of interest to Steel-Belted Radius. Only these attributes are placed in the Variable Table.

Each [Attributes/*name*] section in the LDAP authentication header file lists LDAP attributes of interest to a specific LDAP Search request. The syntax is as follows:

```
[Attributes/name]
attribute
attribute
.
.
.
```

where *attribute* is the name of an LDAP attribute and *name* is an arbitrary name for the section. You must type the *attribute* names exactly as they appear in your LDAP database schema. Use one line per attribute. For example:

```
[Attributes/InterestingAttributes]
User-Secret
RADIUS-Profile
Inactivity-Timeout
```

An [Attributes/*name*] section is associated with a Search request by referencing it from within a [Search/*name*] section using the Attributes field. For example:

```
[Search/DoLdapSearch]
Attributes = InterestingAttributes
```

If the Attributes field is omitted from a [Search/*name*] section, Steel-Belted Radius retains all of the attributes associated with the LDAP entry. Of these attributes, Steel-Belted Radius uses only those referenced in the .aut header file; all

others stay in the Variable Table until the authentication transaction is complete and the table is discarded.

For BindName authentication, you must ensure that the [Attributes/*name*] section lists the attribute in which the user's password is stored and that your [Response] section assigns the value of this attribute to the outgoing %Password parameter. Steel-Belted Radius completes authentication by comparing the returned %Password value with the password that arrived in the Access-Request. For example:

```
[Attributes/InterestingAttributes]
User-Secret
RADIUS-Profile
Inactivity-Timeout

[Response]
%Password = User-Secret
%Profile = RADIUS-Profile
Vendor-Specific-NAS-Attribute = Inactivity-Timeout
```

LDAP Authentication [Search/*name*] Sections

Each [Search/*name*] section in the LDAP authentication header file specifies the complete details of one LDAP Search request. You can use the same Search request on various databases, because the details of the database connection are specified separately.

See “LDAP Authentication [Server/*name*] Sections” on page 421.

For BindName authentication, you must ensure that each [Search/*name*] section searches for a database entry that matches the incoming username and retrieves from it an attribute containing that user's password. Steel-Belted Radius must compare this password to the one it received in the incoming Access-Request packet.

A [Search/*name*] section may retrieve other LDAP attributes as well; however, if you are authenticating with BindName, the user's password is a minimum requirement. Use the Attributes parameter to specify the list of items you want returned.

For example:

```
[Search/DoLDAPSearch]
Base = ou=Special Users, o=bigco.com
Scope = 1
Filter = uid=<User-Name>
Attributes = InterestingAttributes
Timeout = 20
%DN = dn
```

```
[Attributes/InterestingAttributes]
User-Secret
RADIUS-Profile
Inactivity-Timeout
```

```
[Response]
%Password = User-Secret
%Profile = RADIUS-Profile
Vendor-Specific-NAS-Attribute = Inactivity-Timeout
```

The following fields may be present in a [Search/*name*] section:

.aut File

[Search/*name*]

Field	Meaning for LDAP Authentication
%DN	Specifies a variable into which the distinguished name that results from the Search should be placed.
Attributes	The value of this field is a string, <i>name</i> . The <i>name</i> specifies the LDAP attributes of interest to Steel-Belted Radius, by referencing an [Attributes/ <i>name</i>] section elsewhere in the same .aut file.
Base	Specifies the distinguished name (DN) of the entry that serves as the starting point for the search. This filter is a template for an LDAP distinguished name string. The filter follows conventional LDAP syntax and may be as simple or as complex as LDAP syntax permits. It may also include replacement variables from the Variable Table. Each replacement variable consists of the variable name enclosed in angle brackets (<>). Upon execution of the LDAP Search request, the value of the variable replaces the variable name.
OnFound	Specifies the next request section when data is found. 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 same .aut file. If there is no next request section, the overall operation succeeds. This can be overridden using the \$reject keyword, which causes the operation to fail when data is found.
OnNotFound	Specifies the next request section when data is not found. 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 same .aut file. If there is no next request section, the overall operation fails. This can be overridden using the \$accept keyword, which causes the operation to succeed when data is not found.
Search	The value of this optional 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 same .aut file. Steel-Belted Radius tries this Search request next, if the current Search yields no result. Note that each [Search/ <i>name</i>] section may contain at most one Search field.

.aut File

[Search/*name*]

Field	Meaning for LDAP Authentication
Filter	<p>Specifies the filter to apply to the search. This filter is a template for an LDAP Search string. The filter follows conventional LDAP syntax and may be as simple or as complex as LDAP syntax permits, with multiple attribute/value assertions in boolean combination. It may also include replacement variables from the Variable Table.</p> <p>Each replacement variable consists of the variable name enclosed in angle brackets (<>). Upon execution of the LDAP Search request, the value of the variable replaces the variable name.</p> <p>For example, a Search template that uses the User-Name and Service-Type attributes from the RADIUS request might look like this:</p> <pre>(&(uid = <User-Name>) (type = <Service-Type>))</pre>
Scope	<p>Specifies the scope of the search; 0 (search the base), 1 (search all entries one level beneath the base), or 2 (search the base and all entries beneath the base at any level).</p>

The Search field can be used in one [Search/*name*] section after another to create a serial “chain” of Search requests. Every Search in the chain is tried. If any Search fails to return data, the Access-Request is rejected.

An example of a two-part chained Search follows:

```
[Settings]
Search = DoLdapSearch

[Search/DoLdapSearch]
Base = . . .
Filter = . . .
Search = GetMoreLdapInfo

[Search/GetMoreLdapInfo]
Base = . . .
Scope = . . .
Filter = . . .
```

Search sequencing is flexible. You can proceed to a new search even if the current search returns no data by using the `OnNotFound` field. You can also override search results using the `$reject` and `$accept` keywords. The following is an example of flexible searching:

```
[Search/DoSearch2]
Base = o=xyz.com
Scope = 2
Filter = uid=<User-Name>
```

```

Attributes = AttrList
Timeout = 20
%DN = dn
OnFound = DoSearch8
OnNotFound = DoSearch9

[Search/DoSearch8]
Base = o=xyz.com
Scope = 2
Filter = uid=<User-Name>
Attributes = AttrList
Timeout = 20
%DN = dn
OnFound = DoSearch9
OnNotFound = DoSearch9

[Search/DoSearch9]
Base = o=xyz.com
Scope = 2
Filter = uid=<User-Name>
Attributes = AttrList
Timeout = 20
%DN = dn
OnNotFound = $accept

```

LDAP Authentication [Request] Section

The [Request] section of the LDAP authentication header file indicates which RADIUS attribute values Steel-Belted Radius extracts from the incoming Access-Request. Steel-Belted Radius places these values in the Variable Table before moving on to the LDAP Bind and Search requests indicated in the file.

The syntax is as follows:

```

[Request]
attribute = variable
attribute = variable
.
.
.

```

where *attribute* is the name of a RADIUS attribute or other special item associated with the incoming Access-Request, and *variable* is the name of a variable in the Variable Table. The end result of the [Request] syntax is that the value in the incoming attribute is assigned to this variable.

attribute may be the name of a RADIUS attribute, or it may be one of the following keywords, which identify various special items also associated with the connection request. Note that each of these keywords begins with the percent sign (%) to strongly distinguish it from the RADIUS attributes.

Item	Meaning for LDAP Authentication
%OriginalUserName	The original full identification of the user, prior to any processing (i.e., <code>user@realm</code>).
%User	The user portion of OriginalUserName (the section before '@').
%UserName	The full user identification (user and realm strings) after all stripping and processing has been performed.
%Name	Synonym for UserName.
%EffectiveUser	The name of the user (the section before '@') as presented to the authentication method (i.e., possibly modified).
%Realm	The realm portion of the original user identification (the section after '@') as presented to the authentication method (i.e., possibly modified).
%EffectiveRealm	The realm portion of the user identification as presented to the method (i.e., possibly modified).
%NASName	The name of the NAS device, as specified in a RAS Clients entry in the Steel-Belted Radius database.
%NASAddress	The address of the NAS device, in dotted notation.
%NASModel	The make/model of the NAS device, as specified in the Steel-Belted Radius database.
%Password	The PAP password.
%AllowedAccessHours	The times that the user is allowed to be logged in.

variable may be omitted from any [Request] entry. If so, the value in the incoming *attribute* is assigned to a variable named *attribute*.

```
[Request]
attribute =
```

In the following [Request] section example, the `nasid` variable receives the value of the NAS-Identifier attribute from the request packet, the `Service-Type` variable receives the value of the Service-Type attribute, and the `%NASAddress` variable receives the NAS address in dotted notation.

```
[Request]
NAS-Identifier = nasid
Service-Type =
%NASAddress =
```


LDAP Authentication [Defaults] Section

The [Defaults] section of the LDAP authentication header file is used to initialize variables at the start of the LDAP authentication transaction. If not overridden, these are the values used when it is time to Bind, Search, or return an Access-Response. Any variable not listed in the [Defaults] section is initialized to a null value.

The format of each [Defaults] entry is:

```
variable = value
```

where *variable* is the name of a variable and *value* is the value you want to assign to it. For example:

```
[Defaults]
Filter = campus_only
SessionLimit = 600
```

The above example sets values for the `Filter` and `SessionLimit` variables. These variables are standard elements of header file syntax. If you set a `Filter` value in the [Defaults] section, you can override this default by providing the `Filter` field in a [Search/*name*] section. If you set a `SessionLimit` value in the [Defaults] section, you can override this default by providing the `SessionLimit` field in a [Server/*name*] section, and so on.

You can use the [Defaults] section to set default values for any variable, including temporary variables and those that represent RADIUS attributes or LDAP attributes. This way, if the Access-Request packet and LDAP database don't provide Steel-Belted Radius with all of the values that it needs to respond to an Access-Request, in each case it has an acceptable alternative value that can be used instead.

You can store multiple values for any variable; and if that variable is mapped to a RADIUS attribute, all values are returned in the RADIUS response. Multiple entries set within this section are considered multiple values of the same variable.

Variable values are not additive between this section and each search. Therefore, if a search returns one or more values, all current values are replaced.

Note: The [Defaults] section is the only section in the header file that allows you to assign static values to variables.

LDAP Authentication [Server/*name*] Sections

Several sections of the LDAP authentication header file work together to configure the connection between the Steel-Belted Radius server and the LDAP database server(s) that are being used to provide external database authentication. The sections are [Server], [Server/*name*], and [Settings].

Each [Server/*name*] section of the LDAP authentication header file contains configuration information about a single LDAP server. You must provide a [Server/*name*] section for each server you've named in the [Server] section. For example:

```
[Server]
s1=
s2=

[Server/s1]
Host = ldap_1
Port = 389
.
.
.

[Server/s2]
Host = 130.4.67.1
LastResort = 1
.
.
.
```

The following fields may be present in a [Server/*name*] section:

.aut File

[Server/*name*]

Field	Meaning for LDAP Authentication
Bind	<p>For Bind authentication, you must specify a Bind template in the [Settings] section of the LDAP authentication header file.</p> <p>The Bind template must follow conventional LDAP syntax. It may be as simple or as complex as LDAP syntax permits, with multiple attribute/value assertions in boolean combination. It may also include replacement variables from the Variable Table.</p> <p>Each replacement variable consists of the variable name enclosed in angle brackets (<>). Upon execution of the LDAP Bind request, the value of the variable replaces the variable name.</p> <p>For example, a Bind template that uses the User-Name attribute from the RADIUS request might look like this:</p> <pre>uid=<User-Name>, ou=Special Users, o=bigco.com</pre>

.aut File**[Server/name]****Field****Meaning for LDAP Authentication**

BindName	<p>For BindName authentication, the BindName field specifies the distinguished name (DN) to be used in the Bind request that connects to the LDAP server. The [Server/name] section allows 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> <p>For Bind authentication, omit all Bind, BindName and BindPassword fields and use the Bind field in the [Settings] section. See “LDAP Authentication [Settings] Section” on page 425.</p>
BindPassword	<p>For BindName authentication, you must provide a BindPassword. The BindPassword specifies the password to be used in the Bind request that connects to the LDAP server. The [Server/name] section allows you to specify a unique BindPassword for a specific server. Use the [Settings] section to specify a default BindPassword to use for all servers.</p> <p>For Bind authentication, omit the BindName and BindPassword fields. Use the Bind field instead.</p>
Certificates	<p>Specifies the path of the certificate database for use with SSL. This path must not end in a filename. The certificate database must be the cert7.db file used by Netscape Communicator 4.x or later.</p>
ConnectTimeout	<p>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 may or may not implement the feature.</p>
FlashReconnect	<p>If the server is down when performing a Bind or a Search, setting this field to 1 triggers a reconnection attempt before rejecting the request. Therefore, requests are not rejected due to inactivity timeouts.</p> <p>This setting applies to a particular server. To apply it for all servers, place it in the [Settings] section.</p>
Host	<p>The host name or IP address of the LDAP server.</p>
LastResort	<p>You may identify a “last resort” LDAP server by providing a LastResort field in one of these [Server/name] sections, and setting its value to 1. If an LDAP query against some other server results in “no record found,” the authentication server tries the last resort server before accepting or rejecting the user.</p> <p>You might use the LastResort field to identify your master accounts database. This enables Steel-Belted Radius to cover the case in which a user account is newly added but has not yet been propagated to all the LDAP databases.</p>
LdapVersion	<p>Specifies the version of LDAP protocol, if needed to override the default given in the [Settings] section.</p>

.aut File	
[Server/name]	
Field	Meaning for LDAP Authentication
MaxConcurrent	Specifies the maximum number of instances of a single LDAP request that may be executing at one time.
MaxWaitReconnect	Specifies the maximum number of seconds to wait after successive failures to reconnect after a failure of the database connection. WaitReconnect specifies the time to wait after failure of the database connection. This value is doubled on each failed attempt to reconnect, up to a maximum of MaxWaitReconnect.
Password	Specifies the password string, which can include variables, used to specify a Bind prior to any search within a request. If this field is not specified, the packet's password is used.
Port	The TCP port of the LDAP server, or 0 to use the standard port. The default is 0.
QueryTimeout	Specifies the number of seconds to wait for the execution of an LDAP request to complete before timing out. This value is passed to the database engine, which may or may not implement the feature.
Search	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 same .aut file.
SSL	Specifies whether to use SSL over the LDAP connection. The choices are: 0 (do not use SSL), 1 (use SSL). The default is 0.
WaitReconnect	Specifies the number of seconds to wait after a failure of the database connection before trying to connect again.

LDAP Authentication [Server] Section

The [Server] section of the LDAP authentication header file lists the LDAP servers that may be used to perform authentication. Optionally, the [Server] section can also be used to specify multiple LDAP servers for load-balancing or backup; Steel-Belted Radius authenticates against these databases in a round-robin fashion.

The syntax is as follows:

```
[Server]
  ServerName=TargetNumber
  ServerName=TargetNumber
  .
  .
  .
```

where *ServerName* is the name of the header file section that contains configuration information for that server, and *TargetNumber* is an *activation target*

number, a number that controls when this server is activated for backup purposes. *TargetNumber* is optional and may be left blank. For example:

```
[Server]
s1 =
s2 =

[Server/s1]
.
.      ;Connection details for server s1
.
[Server/s2]
.
.      ;Connection details for server s2
.
```

A Steel-Belted Radius server maintains connectivity with its LDAP 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 LDAP servers that Steel-Belted Radius is connected to 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. An activation target of **0** indicates that, in the current configuration, this machine is never used.

LDAP Authentication [Settings] Section

The [Settings] section of the LDAP authentication header file forms a basis for all Bind and Search requests to the LDAP database server(s).

Search sequencing is flexible. You can proceed to a new search even if the current search returns no data by using the `OnNotFound` field. You can also override search results using the `$reject` and `$accept` keywords.

For examples of using flexible searching, see “LDAP Authentication [Server/name] Sections” on page 421.

The fields in the [Settings] section apply to all LDAP servers listed in the header file. The following fields are usually present. If any of these fields is not provided in the [Settings] section, the field assumes a system default value.

.aut File	
[Settings] Field	Meaning for LDAP Authentication
Bind	<p>For Bind authentication, you must specify a Bind template in the [Settings] section of the LDAP authentication header file.</p> <p>The Bind template must follow conventional LDAP syntax. It may be as simple or as complex as LDAP syntax permits, with multiple attribute/value assertions in boolean combination. It may also include replacement variables from the Variable Table.</p> <p>Each replacement variable consists of the variable name enclosed in angle brackets (<>). Upon execution of the LDAP Bind request, the value of the variable replaces the variable name.</p> <p>For example, a Bind template that uses the User-Name attribute from the RADIUS request might look like this:</p> <pre>uid=<User-Name>, ou=Special Users, o=bigco.com</pre>
BindName	<p>For BindName authentication, you must omit the Bind field from the LDAP authentication header file. Use the BindName and BindPassword fields instead.</p> <p>In the [Settings] section, BindName and BindPassword specify a default LDAP Bind template to use for all servers. You can also use BindName and BindPassword in [Server/<i>name</i>] sections to override this default for an individual server</p> <p>See “LDAP Authentication [Server/<i>name</i>] Sections” on page 421.</p>
FlashReconnect	<p>If a server is down when performing a Bind or a Search, setting this field to 1 triggers a reconnection attempt before rejecting the request. Therefore, requests are not rejected due to inactivity timeouts.</p> <p>This setting applies to all servers. To apply it for a particular server, place it in the appropriate [Server/<i>name</i>] section.</p>
LdapVersion	<p>Specifies the version of LDAP protocol. Default is 2.</p>
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 may 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>

.aut File

[Settings] Field Meaning for LDAP Authentication

OnFound	Specifies the next request section when data is found. 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 same .aut file. If there is no next request section, the overall operation succeeds. This can be overridden using the <code>\$reject</code> keyword, which causes the operation to fail when data is found.
OnNotFound	Specifies the next request section when data is not found. 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 same .aut file. If there is no next request section, the overall operation fails. This can be overridden using the <code>\$accept</code> keyword, which causes the operation to succeed when data is not found.
Password	Specifies the password string, which can include variables, used to specify a Bind prior to any search within a request. If this field is not specified, the packet's password is used.
PasswordFormat	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. 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 <code>auto</code> . See "LDAP Authentication and Password Format" on page 410.
PasswordCase	If set to <code>U</code> or <code>Upper</code> , the password returned from the LDAP database is converted to uppercase before authentication. If <code>L</code> or <code>Lower</code> , the password is converted to lowercase. If <code>O</code> or <code>Original</code> (the default), the password is not altered before authentication.
Search	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 same .aut file.
Timeout	Specifies the overall timeout for each request, in seconds. The Timeout parameter is distinguished from the <code>QueryTimeout</code> parameter. <code>QueryTimeout</code> is the timeout for each individual search performed against the LDAP server. <code>Timeout</code> is the overall timeout for the entire authentication, comprising the delay in acquiring resources, attempts against multiple LDAP servers, and so forth. Default is 20 seconds.
UpperCaseName	Specifies whether the username should be converted to uppercase. Choices are: 0 (preserve the case of the username), 1 (convert username to uppercase). The default is 0.
UTC	This entry should be set to 0 to show time information in local time, or 1 to show time information in universal time coordinates (UTC).

In addition to these fields, you may also use the [Settings] section to set defaults for many fields that are usually present in a [Server/*name*] section (Bind, BindName, BindPassword, Certificates, ConnectTimeout, Host, MaxConcurrent, MaxWaitReconnect, Port, QueryTimeout, Search, SSL, and WaitReconnect). The value set in [Settings] provides a default that applies to all servers. This default can be overridden for a particular server by entering the same field with a different value in any [Server/*name*] section.

LDAP Authentication [Failure] Section

The [Failure] section of the LDAP authentication header file can be used to determine the result of the authentication process (accept or reject) when connectivity to all of the configured LDAP databases has failed. For example:

```
[Failure]
Accept = 1
Profile = XYZ
FullName = Mr Stanley Smith
```

The following fields may be present:

Note: The Profile option and the Alias option cannot be used together. Read the descriptions below and choose the one that suits your needs.

.aut File	
[Failure] Field	Meaning
Accept	If Accept is set to 1, Steel-Belted Radius returns an Access-Accept packet with the Profile, FullName, and/or Alias attributes specified in the corresponding [Failure] section fields. If Accept is set to 0, the user is rejected.
Profile	This is the name of an existing Steel-Belted Radius Profile entry, whose Check-List and Return-List attributes are applied to the user's connection.
FullName	This string is the full user name, which is used in the Class attribute in the Access-Accept message.
Alias	As an alternative to using the Profile parameter, you can use the Alias parameter to name an existing Steel-Belted Radius Native User entry. Steel-Belted Radius then applies the Check-List and Return-List attributes of this User entry to the user's connection. <i>NOTE: The Alias feature permits the Concurrent connection limit (settable in the Users dialog, but not in the Profiles dialog) to be applied to the user's connection.</i> See "Concurrent User Connections" on page 79.

.aut File**[Failure] Field Meaning**

Important: You are strongly recommended to use *Profile*, as use of *Alias* has been deprecated. The *LoginLimit* value allows you to implement the concurrent connection limits previously available through *Alias*.

If you want to apply concurrent connection limits to users who are being authenticated via LDAP, you must set up a Native User entry specifically for this purpose, with all of the appropriate Check-List and Return-List attributes, and with no password. You can set up as many such accounts as you require. These entries store a specific set of Check-List and Return-List attributes for LDAP authentication, for use only with the *Alias* parameter.

NOTE: Native User entries without passwords automatically cannot be authenticated. This is a safety feature built into Steel-Belted Radius. Therefore, setting up User entries in preparation for using the *Alias* parameter with LDAP authentication does not pose a “back door” security risk.

NOTE: The Native User authentication method displayed in the Configuration dialog does not need to be activated for the *Alias* feature to work.

LDAP Authentication [Bootstrap] Section

The [Bootstrap] section of the LDAP authentication header file specifies information that Steel-Belted Radius uses to load and start the LDAP Authentication plug-in.

After you edit `ldapauth.aut` and restart Steel-Belted Radius, the `InitializationString` value that you entered in the [Bootstrap] section of `ldapauth.aut` appears in the Configuration dialog’s Authentication Methods list. You can then enable, disable, or prioritize this method just like any other entry in the list.

See “Configuring the Authentication Sequence” on page 38.

You can configure more than one LDAP authentication method. Each requires its own `.aut` file in the same directory as `ldapauth.aut`. The [Bootstrap] section of each `.aut` file must provide a `LibraryName` of `ldapauth.so` (for UNIX) or `ldapauth.dll` (for Windows). The `InitializationString` in each `.aut` file must be unique, so that you can distinguish between authentication methods in the Configuration dialog.

The [Bootstrap] section may contain the following fields.

.aut File	
[Bootstrap] Field	Meaning for LDAP Authentication
LibraryName	This entry must be set to the name of the LDAP authentication module (ldapauth.so or ldapauth.dll).
Enable	This entry must contain a 1 to enable the module, 0 to disable it. If disabled, the authentication method is unavailable and does not appear in the Configuration dialog's Authentication Methods list.
InitializationString	This entry is used to specify the name of the authentication method to appear in the Configuration dialog's Authentication Methods list. In the original header file (ldapauth.aut), this entry is set to "LDAP". You may alter this name if you want. The name of each authentication method must be unique. If you create additional .aut files to implement authentication against multiple databases, be sure that each InitializationString is set to a different method name.

LDAP Authentication Sequence

The sequence of an LDAP authentication transaction is controlled by the LDAP authentication header file as follows:

- 1 The Variable Table is initialized to default values as specified in the [Defaults] section. All variables that are not listed in the [Defaults] section are initialized to null values.
- 2 The values of RADIUS attributes in the Access-Request are copied to the Variable Table, as specified in the [Request] section.
- 3 If a Bind entry was specified in the [Settings] section, authentication via LDAP Bind is now performed. The Bind entry is used as a template to construct a bind string, using replacement values from the Variable Table. An LDAP Bind is then performed to authenticate the user.
- 4 An LDAP Search request is performed for each [Search/*name*] section specified. You may specify zero or more separate Search requests.

For each Search request, LDAP Base and Filter strings are constructed from templates, using replacement values from the Variable Table. These Base and Filter strings are then transmitted to the LDAP server in a Search request.

Each attribute/value pair returned by the LDAP Search is used to set the value of the corresponding entry in the Variable Table. Also, the DN returned by the search may be used to set a variable.

- 5 If a %Password entry appears in the [Response] section, authentication is now performed. The password entered by the user is validated against the value that appears in the %Password variable, and the user is rejected if the passwords don't match.
- 6 If a %Profile entry appears in the [Response] section, the value of the %Profile variable is used to look up a Profile entry in the Steel-Belted Radius database. The Check-List and Return-List attributes in that Profile are used to validate the request and return an appropriate response.
- 7 If a %Alias entry appears in the [Response] section, the value of the %Alias variable is used to look up a Native User entry in the Steel-Belted Radius database. The current transaction is treated as if it came from the "alias" user; that is, the Check-List and Return-List attributes of the alias user are used to validate the request and return an appropriate response.
- 8 If neither a %Profile nor a %Alias entry appears in the [Response] section, then RADIUS attributes for the response packet are created from the Variable Table, based on attribute entries in the [Response] section.

LDAP Authentication Examples

This topic provides examples of LDAP authentication header file syntax. The examples illustrate how you might:

- Authenticate passwords (Bind or BindName).
- Specify Check-List and Return-List attributes (list the attributes or name a profile entry in the Steel-Belted Radius database).

Bind Authentication with Default Profile

The following example is one of the simplest possible LDAP authentication header files. Every user is authenticated using a Bind request to the LDAP database. The same Steel-Belted Radius attribute profile is applied to every Access-Request.

```
[Settings]
MaxConcurrent=1
Timeout=20
ConnectTimeout=25
QueryTimeout=10
```

```

WaitReconnect=2
MaxWaitReconnect=360
Bind=uid=<User-Name>, ou=Special Users, o=bigco.com
LogLevel = 2
UpperCaseName = 0
PasswordCase=original
SSL = 0

[Server]
s1=

[Server/s1]
Host=199.185.162.147
Port = 389

[Defaults]
TheUserProfile = Sample

[Request]
%User-Name = User-Name

[Response]
%Profile = TheUserProfile

[Search/DoLdapSearch]
Base = ou=Special Users, o=bigco.com
Scope = 2
Filter = uid=<dialup>
Attributes = AttrList
Timeout = 20
%DN = dn

[Attributes/AttrList]

```

This example could be even simpler, in that the [Response] section could be empty. If so, Steel-Belted Radius would pass the Bind results (accept or reject) directly to its client and no additional RADIUS attributes would be returned in the Access-Response.

The above example follows a standard Netscape schema that you may already be using on your network. If you substitute your domain name for `bigco.com`, and use the Steel-Belted Radius Administrator to create a profile called `Sample`, you may be able to make this sample header file work in your own test environment.

Note that this example provides empty [Search/*name*] and [Attributes/*name*] sections. It's a good idea not to delete section headings from the file. However, the sections themselves may be empty or unused, as shown here.

BindName Authentication with Callback Number Returned

In the following example, requests are authenticated using Search. BindName and BindPassword values are supplied to permit a connection to the LDAP database. Return-List attributes for authentication are listed in the [Response] section. In this example, the NAS device needs a callback number to complete the connection. The value of the incoming DNIS attribute Calling-Station-ID is used to ensure that the callback number is the number from which the user's request originated.

Note: This example is incomplete; it omits the [Bootstrap] and [Settings] sections to save space.

```
[Server]
s1=

[Server/s1]
Host = 67.186.4.3
Port = 389
BindName=uid=admin, ou=Administrators,
ou=TopologyManagement, o=NetscapeRoot
BindPassword=ourlittlesecret
Search = DoLdapSearch

[Defaults]
SendThis = DidLDAPAuthSearch

[Request]
%UserName = dialup
Calling-Station-ID = thenumbertocall

[Search/DoLdapSearch]
Base = ou=Special Users, o=bigco.com
Scope = 2
Filter = uid=<dialup>
Attributes = AttrList
Timeout = 20
%DN = dn

[Attributes/AttrList]
dialuppassword

[Response]
>Password = dialuppassword
Reply-Message = SendThis
Ascend-Callback-No = thenumbertocall
```

LDAP Bind with Profile Based on NAS Device

In the following example, requests are authenticated using Bind. Check-List and Return-List attributes for authentication are provided by referencing a profile entry in the Steel-Belted Radius database. The profile to be used depends on the specific NAS device from which the user's request originates. Steel-Belted Radius retrieves the profile name by the LDAP database for an IP address that matches the address of the requesting NAS. If this search fails, a profile called `limited` is used. If a profile name is successfully retrieved from the LDAP database, but no profile by that name can be found in the Steel-Belted Radius database, authentication fails due to "lack of resources" and the user is rejected.

Note: This example is incomplete; it omits the [Bootstrap] section and many [Settings] entries to save space.

```
[Settings]
Bind=uid=<loginID>, ou=Special Users, o=bigco.com
Search = DoLdapSearch
```

```
[Server]
s1=
```

```
[Server/s1]
Host = 67.186.4.3
Port = 389
```

```
[Request]
%UserName = loginID
%NASAddress = deviceIP
```

```
[Defaults]
%Profile = limited
```

```
[Search/DoLdapSearch]
Base = ou=CommServers, o=bigco.com
Scope = 1
Filter = ipaddr=<deviceIP>
Attributes = AttrList
Timeout = 20
%DN = dn
```

```
[Attributes/AttrList]
profile
```

```
[Response]
%Profile = profile
```

Quick Reference

A

- [When to Stop and Restart the Server](#)
- [Configuration Files by Feature](#)
- [Configuration Files by Name and Extension](#)

When to Stop and Restart the Server

The **least** drastic action that causes this change to take effect is indicated by **Yes** in this table:

Item changes:	Save the dialog or file	Issue a HUP signal	Stop/restart the server
Access dialog or object	Yes	(Also works)	(Also works)
access.ini file	No	No	Yes
*.acc files	No	No	Yes
account.ini file	No	No	Yes
admin.ini file	No	No	Yes
*.aut files	No	(Sometimes)	Yes
blacklist.ini file	No	No	Yes
bounce.ini file (Windows only)	No	No	Yes
Configuration dialog or object	Yes	(Also works)	(Also works)
*.dct files	No	No	Yes
*.dhc files	No	No	Yes
dhcp.ini file	No	No	Yes
*.dir files (see Notes below)	No	(Sometimes)	Yes
*.eap files	No	(Sometimes)	Yes
eap.ini file	No	No	Yes
enterprises.oid file (UNIX only)	No	No	Yes
events.ini file	No	No	Yes
filter.ini file	No	Yes	(Also works)
Import *.rif or users file	Yes	(Also works)	(Also works)
*.ini for directed accounting	No	No	Yes
IP Pools dialog or object	Yes	(Also works)	(Also works)
IPX Pools dialog or object	Yes	(Also works)	(Also works)
lockout.ini file	No	No	Yes
Log levels (in radius.ini file)	No	Yes	(Also works)
Profiles dialog or object	Yes	(Also works)	(Also works)
Proxy dialog or object	Yes	(Also works)	(Also works)
*.pro files	No	Yes	(Also works)
proxy.ini file (see Notes below)	No	(Sometimes)	Yes
radius.dct file (see Notes below)	No	No	Yes

Item changes:	Save the dialog or file	Issue a HUP signal	Stop/restart the server
radius.ini file	No	No	Yes
RAS Clients dialog or object	Yes	(Also works)	(Also works)
Servers dialog or object	Yes	(Also works)	(Also works)
services file	No	No	Yes
snmpdx.acl file (UNIX only)	No	No	Yes
tacplus.ini file	No	No	Yes
Trace levels	No	Yes	(Also works)
Tunnels dialog or object	Yes	(Also works)	(Also works)
Users dialog or object	Yes	(Also works)	(Also works)
vendor.ini file	No	No	Yes

How to Update Realm Configuration

The following information explains when a HUP signal (RADHUP.EXE under Windows) is sufficient, or insufficient, for updating realm configuration:

- A HUP signal is sufficient to load any changes that you make to proxy.ini, filter.ini, or *.pro files for the purpose of configuring Proxy RADIUS realms.
- However, when you configure directed realms (proxy.ini, *.dir files, and possibly *.acc, *.aut, and accounting *.ini files as well) you must load configuration changes as follows. If you have added or changed:
 - Any directed accounting methods at all, you must stop and restart the server.
 - Directed authentication methods in which external database (SQL or LDAP) authentication is used, you must stop and restart the server.
 - Directed authentication methods in which local or pass-through (Native, UNIX, Domain, Host, SecurID, or TACACS+) authentication is used, a HUP signal is sufficient.

Configuration Files by Feature

Note that some of the files listed below are specific to one operating system. Files are marked with (UNIX) if they exist only under UNIX versions of the package or (Windows) if they exist only under Windows versions of the package.

Feature	Configuration File	File Purpose	Page
Access levels	access.ini	Define access levels	177
	admin.ini	Assign access levels to administrative accounts	184
Account lockout	lockout.ini	General configuration	207
Accounting	account.ini	General configuration	178
	yyyymmdd.ACT	Log file with daily rollover at midnight	148
	yyyymmdd_hhmm_nnnn.ACT	Log file with configured rollover times	181
Administrator (UNIX)	default.htm	Launch the Administrator applet from your browser	84
	index.htm	Alternative means of launching the Administrator applet from your browser	84
Administrator (Windows)	radadnt.exe	Administrator program executable file	84
Attribute editing	filter.ini	Configure attribute editing	202
Attribute Value Pooling	*.rr	To define the Attribute Value sets for this pool and the weights of each set.	247
Auto-restart (UNIX)	radiusd	Script that supports the auto-restart feature by executing radius as an undaemonized child process	250
Auto-restart (Windows)	bounce.ini	General configuration	192
Blacklisting	blacklist.ini	General configuration	191
Dictionaries	vendor.ini	Map vendor-specific dictionary files to identifiers used in the server's administrative database	235
	dictiona.dcm	Keep master list of dictionary files	239
	*.dct	Vendor-specific dictionary files for various NAS devices	239
	radius.dct	Standard RADIUS dictionary file	239
Directed authentication and	proxy.ini	General realm configuration	268
Directed accounting	*.dir	Configure directed authentication and directed accounting realms	292

Feature	Configuration File	File Purpose	Page
DHCP support	dhcp.ini	General configuration	194
	*.dhc	Pool configuration	196
Documentation	readme.txt	Provide late-breaking information not found in the manual	
EAP	eap.ini	Configure operation of EAP	306
Events and counters	perfmon.exe (Windows)	Performance Monitor executable file	163
	events.ini	General configuration	200
External databases	*.acc	Each .acc file configures a SQL accounting method	395
	*.aut	Each .aut file configures a SQL or LDAP authentication method	373 412
Import / Export	*.dci	<p>Dictionaries for importing users files that include vendor-specific attributes</p> <p><i>NOTE: This type of file is not created by Steel-Belted Radius, but the data contained within a users file may be imported from another RADIUS server.</i></p>	239
	annex.dci	Dictionary for importing users files that include Annex vendor-specific attributes	141
	ascend.dci	Dictionary for importing users files that include Ascend vendor-specific attributes	141
	portsmstr.dci	Dictionary for importing users files that include Portmaster vendor-specific attributes	141
	*.rif	<p>RADIUS Information file</p> <p><i>NOTE: This is the format that Steel-Belted Radius uses when exporting user data to a file, and the default file format that it uses when importing user data from another RADIUS server.</i></p>	139
	users	<p>A specially formatted text file, usually provided by RADIUS implementations based on source code from Livingston and Ascend, which contains user data</p> <p><i>NOTE: This type of file is not created by Steel-Belted Radius, but the data contained within a users file may be imported from another RADIUS server.</i></p>	141
	Installation (UNIX)	install.sh	Server installation and configuration

Feature	Configuration File	File Purpose	Page
LDAP Configuration Interface	*.ldif	LDIF-format input files	348
Proxy RADIUS	proxy.ini	General realm configuration	268
	*.pro	Configure Proxy RADIUS realms	278
	filter.ini	Configure attribute editing	202
RADIUS	rfc2865.txt	Documentation of official authentication standards	32
	rfc2866.txt	Documentation of official accounting standards <i>NOTE: These RADIUS standard documents are RFC 2865 and 2866.</i>	32
RADIUS server (UNIX)	default.htm	Launch the Administrator applet from your browser	84
	index.htm	Alternate means of launching the Administrator applet from your browser	84
	install.sh	Server installation and configuration	6
RADIUS server	yyyymmdd.LOG	Server activity log file with daily rollover at midnight	144
	radadnt.exe (Windows)	Administrator program	84
	radius (UNIX)	Server daemon	12
	radius.exe (Windows)	Server executable	17
	radius.ini	General configuration	208
	radiusd (UNIX)	Script that supports the auto-restart feature by executing radius as an undaemonized child process	250
	resetpwd (UNIX)	Reset a forgotten server password	132
	services	UDP port settings <i>NOTE: This file may be found in UNIX under /etc/services and in Windows under C:\winnt\system32\drivers\etc\services</i>	246
Realms	proxy.ini	General configuration	268
	*.pro	Configure Proxy RADIUS realms	278
	filter.ini	Configure attribute editing	202
	*.dir	Configure directed authentication and directed accounting realms	292
Reporting (Windows)	REPORT.RTF	Default output filename and file type (Rich Text Format)	161

Feature	Configuration File	File Purpose	Page
SecurID authentication	sdconf.rec	General configuration <i>NOTE: This is a Security Dynamics file that must be copied to C:\winnt\system32 under Windows or the server directory under Windows as part of configuring Steel-Belted Radius for authentication.</i>	24
Service type mapping	servtype.ini	General configuration	454
SNMP support (UNIX)	enterprises.oid	SNMP object identifier	22
	fnkradtr.mib	Management Information Base for RADIUS server traps and alarms	326
	radsnmp	RADIUS server SNMP Sub Agent daemon	325
	radsnmp.acl	SNMP access control list file	328
	radsnmp.inf	TCP port configuration for SNMP	325
	*.reg	Register SNMP Sub Agents	325
	radsnmp.reg	Register the RADIUS server's SNMP Sub Agent	325
	radsnmp.rsrc	SNMP resource file	325
	snmpdx	Solstice Enterprise Agent (SEA) Master Agent daemon	325
	snmpdx.acl	SNMP traps access control list file	327
		<i>NOTE: SNMP configuration files that are not in the server directory or its subdirectories may be found in either /etc/snmp/conf or /usr/lib/snmp</i>	22
TACACS+ authentication	tacplus.ini	Define connection to TACACS+ server	231
Vendor-specific attributes	vendor.ini	Map vendor-specific dictionary files to identifiers used in the server's administrative database	235
	dictiona.dcm	Keep master list of dictionary files	241
	*.dct	Vendor-specific dictionary files for various NAS devices	239
	radius.dct	RADIUS standard dictionary file	53

Configuration Files by Name and Extension

Note that some of the files listed below are specific to one operating system. Files are marked with (UNIX) if they exist only under UNIX versions of the package or (Windows) if they exist only under Windows versions of the package.

Configuration File	Feature	File Purpose	Page
*.acc	External databases	Each .acc file configures a SQL accounting method	395
access.ini	Access levels	Define access levels	177
account.ini	Accounting	General configuration	178
*.acl (UNIX)	SNMP support	Access control list file	328
yyyymmdd.ACT	Accounting	Log file with daily rollover at midnight	148
yyyymmdd_hhmm_nnnn.ACT	Accounting	Log file with configured rollover times	181
admin.ini	Access levels	Assign access levels to administrative accounts	184
annex.dci	Import / Export	Dictionary for importing users files that include Annex vendor-specific attributes	141
ascend.dci	Import / Export	Dictionary for importing users files that include Ascend vendor-specific attributes	141
*.aut	External databases	Each .aut file configures an authentication method such as SQL, LDAP, or EAP-TTLS.	373 412 308
blacklist.ini	Blacklisting	General configuration	191
bounce.ini (Windows)	Auto-restart	General configuration	192
*.dci	Import / Export	Dictionaries for importing users files that include vendor-specific attributes <i>NOTE: This type of file is not created by Steel-Belted Radius, but the data contained within a users file may be imported from another RADIUS server.</i>	141
*.dcm	Dictionaries	Keep master list of dictionary files	239
*.dct	Dictionaries	Vendor-specific dictionary files for various NAS devices	239
default.htm (UNIX)	RADIUS server	Launch the Administrator applet from your browser	84
*.dhc	DHCP support	Pool configuration	196
dhcp.ini	DHCP support	General configuration	194
dictiona.dcm	Dictionaries	Keep master list of dictionary files	239

Configuration File	Feature	File Purpose	Page
*.dir	Directed authentication, directed accounting	Configure directed authentication and directed accounting realms	292
eap.ini	EAP	Configure operation of EAP	306
enterprises.oid (UNIX)	SNMP support	SNMP object identifier file	22
events.ini	Events and counters	General configuration	200
*.exe (Windows)	(Various)	Executable files for: NT Performance monitor (perfmon.exe) Administrator program (radadnt.exe) Server (radius.exe)	163 84 17
filter.ini	Proxy RADIUS	Configure attribute editing	202
fnkradtr.mib (UNIX)	SNMP support	Management Information Base for RADIUS server traps and alarms	326
*.htm, *html (UNIX)	(Various)	Launch the Administrator applet from your browser	84
index.html	RADIUS server	Launch the Administrator applet from your browser	84
*.inf	SNMP support	TCP port configuration	325
*.ini	(Various)	Initialization files for various purposes: access levels (access.ini), accounting attributes (account.ini), administrative accounts (admin.ini), attribute editing (filter.ini), auto-restart (bounce.ini), dictionary integration (vendor.ini), events and counters (events.ini), Proxy Radius or directed realm configuration (proxy.ini), RADIUS server configuration (radius.ini), and TACACS+ authentication (tacplus.ini)	176
install.sh (UNIX)	RADIUS server	Server installation and configuration	6
*.ldif	LDAP configuration interface	LDIF-format input files	348
lockout.ini	Account lockout	General configuration	207
yyyymmdd.LOG	RADIUS server	Server activity log file with daily rollover at midnight	144
*.mib (UNIX)	SNMP support	Management Information Base files specifying support for RADIUS accounting servers (raccs.mib), RADIUS authentication servers (rauths.mib), and SNMP traps and alarms	324

Configuration File	Feature	File Purpose	Page
*.oid (UNIX)	SNMP support	SNMP object identifier file	22
perfmon.exe (Windows)	Events and counters	Performance Monitor executable file	163
portmstr.dci	Import / Export	Dictionary for importing users files that include Portmaster vendor-specific attributes	141
*.pro	Proxy RADIUS	Configure Proxy RADIUS realms	278
proxy.ini	All realms	General realm configuration	268
proxyrl.ini	Smart Static Accounting	Configures list of realms for forwarding accounting packets	277
radadnt.exe (Windows)	RADIUS server	Administrator program executable file	84
radius (UNIX)	RADIUS server	Server daemon	12
radius.dct	Dictionaries	Standard RADIUS dictionary file	239
radius.exe (Windows)	RADIUS server	Server executable	17
radius.ini	RADIUS server	General configuration	208
radiusd (UNIX)	Auto-restart	Script that supports the auto-restart feature by executing radius as an undaemonized child process	250
radsnmp (UNIX)	SNMP support	RADIUS server SNMP Sub Agent daemon <i>NOTE: This file is in the server directory and in /usr/lib/snmp.</i>	325
radsnmp.acl (UNIX)	SNMP support	SNMP access control list file <i>NOTE: This file is in the directory /etc/snmp/conf</i>	328
radsnmp.inf (UNIX)	SNMP support	TCP port configuration <i>NOTE: This file is in the server directory</i>	325
radsnmp.reg (UNIX)	SNMP support	Register the RADIUS server's SNMP Sub Agent <i>NOTE: This file is in the directory /etc/snmp/conf</i>	325
radsnmp.rsrc (UNIX)	SNMP support	SNMP resource file <i>NOTE: This file is in the directory /etc/snmp/conf</i>	325
readme.txt	Documentation	Provide late-breaking information not found in the manual	
.rec	SecurID authentication	General configuration	24
*.reg (UNIX)	SNMP support	Register SNMP Sub Agents	325
REPORT.RTF (Windows)	Reporting	Default output filename and file type (Rich Text Format)	161

Configuration File	Feature	File Purpose	Page
resetpwd (UNIX)	RADIUS server	Reset a forgotten server password	132
rfc2865.txt	RADIUS	Authentication standard	32
rfc2866.txt	RADIUS	Accounting standard	32
*.rif	Import / Export	RADIUS Information File <i>NOTE: This is the format that Steel-Belted Radius uses when exporting user data to a file, and the default file format that it uses when importing user data from another RADIUS server.</i>	139
*.rr	Attribute Value Pooling	To define a round-robin Attribute Value Pool.	247
*.rsrc (UNIX)	SNMP support	Resource file	325
*.rtf (Windows)	Reporting	Default output filename and file type (Rich Text Format)	161
sdconf.rec	SecurID authentication	General configuration <i>NOTE: This is a Security Dynamics file that must be copied to C:\winnt\system32 (under Windows) or the server directory (under UNIX) as part of configuring Steel-Belted Radius for SecurID authentication.</i>	24
services	RADIUS server	UDP port settings <i>NOTE: This file is in the directory /etc/services (UNIX) or C:\winnt\system32\driver (Windows).</i>	246
servtype.ini	Service type mapping	General configuration	454
snmpdx (UNIX)	SNMP support	Solstice Enterprise Agent (SEA) Master Agent daemon	325
snmpdx.acl (UNIX)	SNMP support	SNMP traps access control list file	328
tacplus.ini	TACACS+ authentication	Define connections to TACACS+ server	231
*.txt	(Various)	Documentation in text format for various purposes, for example late-breaking information about the Steel-Belted Radius product (readme.txt) <i>NOTE: The RADIUS standard documents for authentication (rfc2865.txt) and for accounting (rfc2866.txt) may be found on the web at: http://www.ietf.org/rfc/rfc2865.txt and http://www.ietf.org/rfc/rfc2866.txt</i>	

Configuration File	Feature	File Purpose	Page
users	Import / Export	A specially formatted text file, usually provided by RADIUS implementations based on source code from Livingston and Ascend, which contains user data. <i>NOTE: This type of file is not created by Steel-Belted Radius, but the data contained within a users file may be imported from another RADIUS server.</i>	141
vendor.ini	Dictionaries	Map vendor-specific dictionary files to identifiers used in the server's administrative database.	235

Steel-Belted Radius Vendor-Specific Attributes

Attribute Name	Purpose
Funk-Allowed-Access-Hours	May be placed in the Check-List for a User or profile entry to control the exact time periods during which a user may be allowed access. Funk-Allowed-Access-Hours is a variable-length string that identifies time periods in a 7-day week of 24-hour days. This string consists of one or more day specifiers (each of which may list one or more days and/or ranges of days) followed by one or more ranges of 24-hour times, in minutes.
Funk-Round-Robin-Group	May be placed in the Return-List for a User or profile entry to dynamically assign an attribute set from an Attribute Value Pool at log-in time. The value of this attribute must be set to the .rr file name which defines the Attribute Value Pool.
Funk-Full-User-Name	<i>Reserved for future use</i>
Funk-Concurrent-Login-Limit	<i>Reserved for future use</i>

Technical Bulletins

B

- LDAP Support for Novell NDS
- Service Type Mapping
- User Name Transform
- Routed Proxy
- CCA Support for 3COM
- Ascend Filter Translation
- Idapauth Extensions
- Ericsson's e-h235 Authentication Protocol
- Uniport Plug-In

LDAP Support for Novell NDS

The Steel-Belted Radius LDAP authentication plug-in contains features to enable greater interoperability with Novell NDS.

If you have configured NDS to limit the number of grace logins available to a user, Steel-Belted Radius can be configured to coordinate with NDS on this feature. Each time a user authenticates, the number of grace logins available is decremented until the account is locked out and needs an administrator to unlock it. A profile is assigned to these users — a profile that overrides their normal profile — when they are being authenticated using a grace login.

The features include:

- **Allowing Expired Accounts:** When enabled, this feature allows users to log in even after their account has expired.

Note: As NDS itself does not check the password, this feature should be used only if the administrator has configured the ProfileForExpiredUsers setting to assign an alternate profile to the user, one that would inform the user of their account status but not allow a usable connection to the network. For example, you can use http redirection to force the connection to a web page with relevant information.

Note: Administrators should contact their NAS vendors to determine the capabilities of their NAS equipment and what attribute-value pairs would be needed to create such a connection.

- **Grace Logins:** When grace logins are limited in NDS, Steel-Belted Radius can be configured to accept or reject a user whose password has expired. This user is said to be in *grace login mode*; the user can also be allowed to log in but is provided with an alternate profile.

Important: *The grace login feature requires NetWare 6.0 or later with eDirectory 8.6 or later.*

- **BindName:** You can use the BindName technique to search the NDS directory for a matching user and, having retrieved the user's DN, apply the Bind technique to authenticate the user's credentials. This combination allows the user to specify their common name (rather than the more cumbersome DN) when requesting authentication.

Note: This feature works only if the NetWare server accepts LDAP Bind requests for users who are in grace login state. Earlier versions of the NetWare server did not support this feature.

Important: *You must configure the NDS server to 'Allow Clear_text passwords' to use these LDAP extensions. You can do this from the ConsoleOne application, in the Properties section of the LDAP group entry for your server.*

Configuration

The [NDS] section has been added to the ldapauth.aut file to configure these features. The fields in this section are as follows:

Field	Usage
Enable	Set to 1 to enable the Novell NDS extensions.
AllowExpiredAccountsForUsers	Set to 1 to allow users to authenticate even after their account has expired. The default is 0. Important: <i>This requires that the Netware server notify Steel-Belted Radius that the user's account has indeed expired when the Netware server is attempting to Bind.</i>
ProfileForExpiredUsers	The name of the profile to assign a user (as an override) if authenticated with an expired account. If you do not provide a value for this setting, the user is accepted with the usual profile and attributes. We recommend that you provide the user with a profile with restricted access.
AllowGraceLoginsForUsers	Set to 1 if users should be allowed to be authenticated in grace login mode. This decrements the grace login counter and reject the user once it has run out. The default is 1.
ProfileForGraceLoginUsers	The name of the profile to assign a user (as an override) if authenticated in grace login mode. If you do not provide a value for this setting, the user is accepted with the usual profile and attributes.

If `Enable` is set to 1, Steel-Belted Radius works as follows:

- If the NDS directory is configured to operate without grace logins:
 - If `AllowExpiredAccountsForUsers` is set to 0 (the default), users with expired passwords are rejected.
 - If `AllowExpiredAccountsForUsers` is set to 1, users with expired accounts are accepted; the attributes returned in the Access-Accept

response are either the attributes normally assigned to the user or, if the `ProfileForExpiredUsers` setting is specified, the attributes specified in that profile. If you enable this feature, we recommend that you configure the `ProfileForExpiredUsers` setting.

- If the NDS directory is configured to operate with grace logins:
 - If `AllowGraceLoginsForUsers` is set to 0, users with correct but about-to-expire passwords are rejected.
 - If `AllowGraceLoginsForUsers` is set to 1 (the default), users with correct but about-to-expire passwords are accepted; the attributes returned in the Access Accept are either the regular attributes assigned to the user or, if `ProfileForGraceLoginUsers` is specified, the attributes specified in that profile.

Sample `ldapauth.aut` file

```
[Bootstrap]
LibraryName=ldapauth.dll
Enable=1
InitializationString=NETWARE6Sample
```

```
[Settings]
MaxConcurrent=1
Timeout=20
ConnectTimeout=25
QueryTimeout=10
WaitReconnect=2
MaxWaitReconnect=360
Search=DoLdapSearch
LogLevel = 2
UpperCaseName = 0
PasswordCase=original
SSL = 0
```

```
[Server]
s1=
```

```
[Server/s1]
Host=192.168.5.110
Port = 389
```

```
[Request]
%UserName = User-Name
```

```
[Response]
```

```
%profile= attThatContainsUserProfile
```

```
[Search/DoLdapSearch]
```

```
;Bind as a privileged user or someone that has the right to
```

```
; search the tree and retrieve the DN of users
```

```
bind=cn=administrator,o=netware6
```

```
Password= support
```

```
Base = o=netware6
```

```
Scope = 2
```

```
Filter = uid=<User-Name>
```

```
%DN = dn
```

```
;if the user is found perform search getprofile
```

```
onfound = GetProfile
```

```
;else reject the user
```

```
onnotfound=$reject
```

```
[Search/GetProfile]
```

```
; bind using the DN retrieved in doLdapSearch
```

```
Bind = <dn>
```

```
;You do not have to supply the password SBR knows to use the  
;one received in the auth request.
```

```
;Setting base to the DN is most efficient
```

```
Base =<DN>
```

```
Scope = 2
```

```
Filter = uid=<User-Name>
```

```
Attributes = AttrList
```

```
[Attributes/AttrList]
```

```
attThatContainsUserProfile
```

```
; Configure NDS-specific functions of LDAP Auth plug-in
```

```
; This makes use of NetWare's capability to allow users
```

```
; who are in grace login mode, or expired to login via ldap
```

```
[NDS]
```

```
Enable=1
```

```
AllowExpiredAccountsForUsers=1
```

```
ProfileForExpiredUsers=Expired
```

```
AllowGraceLoginsForUsers=1
```

```
ProfileForGraceLoginUsers=Grace
```

Service Type Mapping

Service type mapping allows a single user to have multiple authorization attribute sets based on the service type the user is requesting. The service type is determined based on request attributes using rules that may differ depending on the NAS device.

Using static configuration parameters in the `servtype.ini` file, you can specify, on a NAS-by-NAS basis, a mapping of request attributes and/or values to service type strings. These strings can be attached to the username, either as a prefix or as a suffix. The elaborated username are used for both authentication and authorization, and for allowing different authorizations based on service type requested.

Configuration

Service type mapping is configured in the following way:

- Create multiple Native User entries in the database according to specific naming and mapping conventions. For example:

```
PPP:George
VPN:George
PPP:Martha
ISDN:Martha
```

- Define a set of rules in the `servtype.ini` file mapping each incoming Access-Request packet to the appropriate database entry for the user.

Native User Database Entries

The Native User entries you define to support service type mapping must follow a consistent naming convention. However, you are free to use any convention you like.

For example, you can store entries for PPP users using the convention `ppp:username` (for example, `ppp:george` and `ppp:martha`) and entries for VPN users using the convention `vpn:username` (`vpn:george` and `vpn:martha`).

For the mapping to work, however, you must define users who do not have any of these mapped prefixes or suffixes in the native users database. For example, if you want to map `vpn:emil` and `ppp:emil` so that the appropriate profiles would be returned, you could enter three user entries in the native users database:

```
vpn:emil
ppp:emil
emil
```


Alternatively, you could omit `emil` from the native users database, authenticate `emil` against a non-native method and then apply the mapping. The mapped names would still have to be in the native user database for profiles to be returned.

You can support classes of service by varying the string you use in creating Native User entries. For example, if you offer three classes of VPN service, your VPN entries might use the conventions `vpn1:username`, `vpn2:username`, and `vpn3:username` (`vpn3:george` and `vpn1:martha`).

A delimiting character (such as a colon) in your service type string makes your user record names easier to read - for example, `vpn:amar` instead of `vpnamar`. When you design a service type string, consider whether it is a prefix (`string+separator`) or a suffix (`separator+string`) to the username.

Note: You can define Native User records using the Administration program or the LDAP configuration interface.

servtype.ini File

The servtype.ini configuration file controls service type mapping and contains the following sections:

servicetype.ini	
Section	Meanings
[Settings]	<p>Indicates how the service type string should be attached to the username prior to look-up in the Native User database: by prefix, by suffix, or not at all. The two fields <code>Prefix</code> and <code>Suffix</code> may be enabled (set to 1) or disabled (set to 0) independently of each other. If both are set to 0 (the default) the service type feature is completely disabled.</p> <p>Using this example, if user <code>george</code> requests PPP service and the string for that service type is <code>ppp:</code>, the Native User record with the Return-List for this request has the name <code>ppp:george</code>.</p>
[NAS]	<p>Allows you to map NAS devices to <code>[mapping]</code> sections. The syntax for [NAS] is as follows:</p> <pre>[NAS] NASname = mapping NASname = mapping . . . NASname = mapping</pre> <p>Each <code>NASname</code> in the [NAS] section must match the name of a RAS Client entry in the database. When an Access-Request is received, its NAS-IP-Address attribute is matched to a RAS Client entry in the database. If a match can be found, and the RAS Client name matches a <code>NASname</code> in the [NAS] section, a corresponding <code>[mapping]</code> section will be found.</p>

servicetype.ini**Section****Meanings**

[mapping]

Each section does the following:

Names the strings to be added to the username for look-ups in the Native User database, to find the correct Return-List.

Provides a set of rules which the incoming Access-Request packet must meet if an Access-Accept is to be returned.

The syntax for each [mapping] section is as follows:

```
[mapping]
```

```
ServiceTypeString
```

```
    RADIUSattribute = value
```

```
    ~RADIUSattribute = value
```

```
    .
```

```
    .
```

```
    .
```

```
ServiceTypeString
```

```
    RADIUSattribute = value
```

```
    RADIUSattribute = value
```

```
    .
```

```
    .
```

```
    .
```

```
    .
```

```
    .
```

```
    .
```

There may be zero or more “rules” in each *ServiceTypeString* section.

Each rule is a statement about an attribute in the incoming Access-Request packet. Each rule begins with a tab character, followed by a *RADIUSattribute=value* string, followed by a carriage return. Every component of the rule is optional, so there are many syntax variations.

If a rule provides a *RADIUSattribute* field, this field must name a standard or vendor-specific RADIUS attribute that is known to the server. If a rule provides an optional *value* field, this field must name a valid possible value for that attribute.

The following logic is applied to the *[mapping]* section:

- 1 The “next” (initially, the “first”) *ServiceTypeString* in the *[mapping]* section is sought. It combines the *ServiceTypeString* with the username as defined in the *[Settings]* section, and tries to find a matching Native User entry.

If a matching entry is found, each rule in the *ServiceTypeString* section is tried against the attributes in the incoming Access-Request packet. Otherwise (there was no next *ServiceTypeString* or no match could be found), the user is rejected.

- 2 In the following example, the rule syntax is:

RADIUSattribute = value

If the *RADIUSattribute* named is present in the Access-Request packet, and if it has the *value* shown, this rule is true. Evaluate the next rule. If there is no next rule, select this *ServiceTypeString*.

If the *RADIUSattribute* named is not present in the Access-Request packet, or if it is present but does not have the value shown, then control returns to step 1.

- 3 In the following example, the rule syntax is:

RADIUSattribute

NOTE: the absence of a value. If the RADIUSattribute is present in the Access-Request packet, this rule is true. Evaluate the next rule. If there is no next rule, select this ServiceTypeString.

If the *RADIUSattribute* is not present in the Access-Request packet, control returns to step 1.

- 4 In the following example, the rule syntax is:

~RADIUSattribute =value

NOTE: the tilde (~) operator. If the RADIUSattribute named is present in the Access-Request packet, and if it does not have the value shown, this rule is true. Evaluate the next rule. If there is no next rule, accept this ServiceTypeString.

If the *RADIUSattribute* named is not present in the Access-Request packet, or if it is present but has the *value* shown, control returns to step 1.

NOTE: the following is not valid syntax:RADIUSattribute = ~value

servicetype.ini**Section****Meanings**

- 5 In the following example, the rule syntax is:
~RADIUSattribute
NOTE: The tilde (~) operator and the absence of a value. If the RADIUSattribute named is not present in the Access-Request packet, this rule is true. Evaluate the next rule. If there is no next rule, accept this ServiceTypeString.
If the *RADIUSattribute* named is present in the Access-Request packet, control returns to step 1.
- 6 If no *RADIUSattribute* rules are provided and a *ServiceTypeString* section exists, but contains no rules, the *ServiceTypeString* is selected automatically.
- 7 If no *ServiceTypeString* sections are provided and a *[mapping]* section exists, but is empty, the user is rejected automatically.
-

In addition to enabling a prefix or suffix, the [Settings] section of the servtype.ini file permits you to specify a default *[mapping]* section to be used when an Access-Request packet arrives from a NAS device that is not listed in the [NAS] section of servtype.ini. The syntax for setting this default is as follows:

```
[Settings]
Default = mapping
```

The Default field is optional. If you do not set up a default mapping, and the server cannot determine the mapping in any other way, the server ignores the service type and authenticates the user without it.

The following is a sample servtype.ini file:

```
[Settings]
Prefix=1
Suffix=0
Default=defaultmap

[NAS]
nas1=nas1map
nas2=nas2map

[nas1map]
ppp:
    Framed-Protocol=1
    Service-Type=2

vpn:
    Framed-Protocol=6
```

```
    ~Service-Type=2

other:
    Framed-Protocol
    Service-Type

[nas2map]
analog:
    NAS-Port-Type=1

isdn:
    NAS-Port-Type=2

[defaultmap]
ppp:
```

To permit simple and clear failure cases, any syntax error in the `servtype.ini` file prevents initialization of the file. If this occurs, service type mapping is disabled. This event is logged in the `date.log` file.

User Name Transform

Steel-Belted Radius allows the administrator to specify a rule for transforming user names from the form in which they are received into a form in which they may be properly processed.

This may be necessary as the form in which users supply their names to the NAS device (and hence to the RADIUS server), is not always compatible with the form which the RADIUS server requires to apply its own rules for proxy forwarding, or with the form which the authentication system (either native or backend NOS, or other authenticator) requires.

Operation

The User Name Transform is a rule used to convert input strings to output strings. The rule is based on two pieces of information:

- 1 Input format
- 2 Output format

This rule is applied to the user name appearing in a RADIUS request. The user name from the RADIUS request is parsed based on the input format. If the user name does not conform to the input format, the rule does not apply and the user name is unchanged.

If the rule does apply, the parsed elements of the user name are formatted based on the output form to construct the transformed user name.

The transformed user name replaces the original user name in processing, just as if the transformed user name had been included in the packet. Therefore, the decision as to whether to proxy forward the packet is based on the transformed user name, and all authentications are based on the transformed user name.

The User Name Transform may be applied to authentication packets, accounting packets, or both. You may want to apply it to both (if at all), for consistency. Alternatively, you may want to apply the transform only to authentication, and base accounting on the user name as it originally appears in the packet.

The Transformation Process

The transformation process operates in the following manner:

- 1 The User-Name from the Access-Accept (or Acct-Start/Acct-Stop) is compared with the input format rule.

- 2 If the User-Name matches the rule, it is modified into the output format, and authentication continues.
- 3 If the User-Name does not match the input format, no modification occurs, and authentication continues.

The Format String

Format strings may be any sequence of characters, and may contain embedded variables enclosed in angle brackets ('<' '>'). The backslash ('\') is an escape character within text, used to represent literal characters. Within variable names, a backslash is treated as a character, not as an escape; and therefore, variable names may not include right angle brackets ('>').

The literal text should be composed of characters not expected to be found in the variable elements. That is, you should use punctuation characters such as a slash ('/') or an at-sign ('@'), rather than letters or numbers.

Configuration

The User Name Transform is configured through the [UserNameTransform] section of the radius.ini. It contains the following fields:

radius.ini	
[UserNameTransform]	
Field	Meaning
In	The input format
Out	The output format
Authentication	Set to <i>Yes</i> to enable the transform for authentication requests. The default value is <i>Yes</i> .
Accounting	Set to <i>Yes</i> to enable the transform for accounting requests. The default value is <i>Yes</i> .
Proxy	Set to <i>Yes</i> to enable the transform for proxied requests. The default value is <i>Yes</i> .

For example, the following settings transforms `george@acme.com` to `george`:

```
In = <user>@<realm>
Out = <user>
```

The following settings transform `abc/martha@bigco.com` to `bigco.com::abc/martha`:

```
In = <prefix>/<user>@<realm>
Out = <realm>::<prefix>/user
```

Routed Proxy

Routed Proxy provides the ability to consult an external database (SQL or LDAP) to determine the routing of an authentication request. This information can be used to pre-authenticate the user, to select a target realm for a subsequent proxy, to modify the User-Name in the proxy request, and to insert attributes into the response.

Normally, a proxy target may be configured statically (Proxy As Authentication Method), or may be determined based on information in the packet, such as NAI (decorated user-name), DNIS, or other computations (attribute mapping). Routed Proxy adds the ability to externalize the dynamic determination of a realm through an LDAP directory or a SQL database.

Typical uses for this feature might be:

- Allowing the User-Name to determine the realm without requiring decoration.
- Centralizing the mapping of attributes to the realm (the attribute mapping feature, but offloaded to LDAP or SQL).
- Allowing the User-Name to be decorated only with a final realm, mapping it to the Next-Hop realm through LDAP. This allows an enterprise to change their ISP without requiring reconfiguration of user PCs. (Also, by storing shared-secret information for the Next-Hop realm in LDAP, Secure Peer Discovery functionality is available.)

Operation

Routed Proxy occurs when certain information is returned from an external database (SQL or LDAP). Operation is governed by the following two variables, accessible by authentication methods through the SQL and LDAP authentication plug-ins:

- If `%ProxyRealm` is not set, Routed Proxy does not occur.
- If `%ProxyRealm` is set to a directed realm, then handle it as a directed realm request.
- If `%ProxyRealm` is set to a proxy realm, then send the request off to that proxy realm.
- `%ProxyUserName` is set to the User-Name attribute, which must be sent in the proxy request. If `%ProxyUserName` is not set, the User-Name from the original request packet is used.

Routed Proxy supports RADIUS authentication, including the RADIUS challenge process and RADIUS accounting. Password authentication may happen when the external database is accessed, or later by the proxy target itself, depending on whether the values of the `%Password` and `%ProxyRealm` variables returned from

the database are blank or non-blank. Note that only one routed proxy is allowed per transaction; that is, they cannot be nested.

The following table lists all four cases for Routed Proxy.

%Password	%ProxyRealm	Action
blank	blank	Authentication fails
blank	non-blank	Authenticate at proxy target
non-blank	blank	Authenticate password now; no proxy
non-blank	non-blank	Authenticate password now; direct to appropriate realm if successful

See “SQL Authentication” on page 364, and “External LDAP Authentication” on page 404.

CCA Support for 3COM

Steel-Belted Radius can support the generation of 3Com CCA tunnel attributes.

Configuration

To enable the return of the required CCA tunnel attributes, the `ccagw.ini` file must be modified.

The `ccagw.ini` file contains information about gateways, which are stored in the `[gateway]` sections. A `[gateway]` section must be present for each gateway supported.

The following table describes each field:

ccagw.ini	
[gateway] Field	Meaning
Address	The address of the gateway.
TunnelRefresh	The number of seconds before the tunnel refreshes. The default value is 0.
Description	A text string describing the gateway.
Secret	The shared secret between Steel-Belted Radius and this gateway device.

For example:

```
[Jupiter-Gateway]
Address = 200.47.98.142
TunnelRefresh = 3600
Description = Jersey City facility, East Coast subscribers
Secret = Holland Tunnel
```

Setting User and Profile Attributes

To enable this functionality for a particular user, the return list for the user must contain the following attributes:

```
Tunnel-Authentication
VPN-Gateway
```

Both of these attributes are defined as strings. The value of each attribute must be set to the name of the gateway used in the `ccagw.ini` file. For example, the return list of a user would have to include:

```
Tunnel-Authentication = Jupiter-Gateway
VPN-Gateway = Jupiter-Gateway
```

It is important to make sure that both attributes name the correct gateway. If an unknown gateway is named, the request is rejected.

Note: Steel-Belted Radius is capable of returning multiple pairs of attributes for different gateways. For each gateway named in one of the attributes, a different random session key is generated.

Note: Please see your 3Com documentation for more information.

Ascend Filter Translation

Ascend defines two attributes — `Ascend-Data-Filter` (242) and `Ascend-Call-Filter` (243) — that contain structured binary data representing a filter to be applied to the NAS device.

Instead of entering hexadecimal strings to configure these attributes, users can configure these attributes as text strings. Steel-Belted Radius automatically converts the text strings to the proper binary representation. The original filter attributes are still supported, and these attributes still may be configured as hexadecimal strings.

The following attributes allow configuration as text:

```
Ascend-Data-Filter-String
Ascend-Call-Filter-String
```

When Steel-Belted Radius formats a response packet, it translates the string version of the attribute to the appropriate binary value, and returns the attribute in the `Access-Accept` message.

Configuration

These attributes may be entered as text strings through the Administrator. The attributes may also be returned from an LDAP or SQL database during authentication.

No syntax validation is performed when the attribute is configured. The validation of syntax occurs only when the response packet is formatted. If the syntax is invalid, a reject response is issued and an error is logged.

Note: These attributes should be tested before configured on a production server.

Two types of filter are supported: “ip” and “generic”. “ipx” filters are not supported.

Syntax

In the syntax descriptions below, brackets ‘[’ ‘]’ indicate that the items enclosed are optional.

```
ip [direction] [action] [srcip address[/mask]] [dstip
address[/mask]] protocol [srcport operator port] [dstport
operator port]
```

Parameter	Values
direction	May be "in" or "out". The default is "out".
action	May be "forward" or "drop". The default is "drop".
address	An IP address in decimal dotted notation.
mask	The number of bits (decimal) in the network portion, from 0 through 32. The default is based on class of network.
protocol	The protocol number (decimal); e.g., 6 for TCP, 17 for UDP. The following protocol names are translated to the proper number: icmp(1), tcp(6), udp(17), ospf(89).
operator	May be "=", "!=", "<", or ">".
port	The port number (decimal). In addition, the following service names are be translated to the proper port number: ftp-data(20), ftp(21), telnet(23), smpt(25), nameserver(42), domain(53), tftp(69), gopher(70), finger(79), www(80), kerberos(88), hostname(101), nntp(119), ntp(123), exec(512), login(513), cmd(514), talk(517).

Example:

```
ip out forward srcip 10.1.1.0/24 6 dstport = 80 srcport < 1023
```

Note: Please see your Ascend documentation for details about the syntax for these attributes.

Idapauth Extensions

This plug-in adds the ability to use two new LDAP attributes:

- `ProfileData`: Stores multiple RADIUS attribute value pairs within a single LDAP container, removing the need for multiple entries in a LDAP user object.
- `GlobalProfile`: Configures users based on a global profile, which can be specified as any user name concatenated with the company name (such as `Profile1@Company`).

GlobalProfile Attribute

The `GlobalProfile` attribute takes the value from an LDAP attribute and parses it to match a profile. The format of the data is that of a DN attribute and should be stored as:

```
cn=profile-name, {optional ou's}, o=name,  
  {optional dc's \o's \c's}
```

profile-name and *name* are concatenated to build `profilename@name`. This value should then match a profile stored in the Steel-Belted Radius server.

For example:

```
cn=Global1, ou=Profile, ou=Radius, ou=IP Services,  
o=funk, o=directoryroot
```

This value is parsed to form a new string: `Global1@funk`. This new string is then passed back as the profile by making the following entry in the response section:

```
[Response]  
%profile= LDAP attribute that contains the global profile
```

This value, however, is ignored if:

- There is no `o` keyword value; *or*
- The string does not begin with the `cn` keyword; *or*
- `%profile` is not set to the name of the attribute that contains the `Globalprofile` data

An incorrect profile name results if the *name* parameter isn't the first value of the organization name (`o`).

ProfileData Attribute

This feature allows an administrator to store multiple RADIUS attribute value pairs within a single LDAP container, removing the need for multiple entries in a LDAP user object.

For example, the values for `framed-ip-address`, `service-type` and `framed-protocol` could all be stored in one attribute called `Std_dialin`. Combining them saves space on the LDAP server.

The attribute should be of a string data-type (directory string or string case insensitive). The format for the data stored in this attribute is:

```
<r|R>;attribute-name;type;value&
```

- `r` or `R` specifies that the attribute may be either single or multi-valued.
- `attribute-name` : Specifies the name of the attribute that is being added.
- `value&` : The value to be returned with this attribute, terminated with `&`

Note: The type field is ignored (the values are interpreted according to the RADIUS dictionary).

For example:

```
Std_dialin: r; service-type; integer; 1&r; framed-protocol;  
integer; 2& r; framed-ip-address; string;192.168.2.2&
```

The `Profiledata` attribute is retrieved from the LDAP server in the same way in which other attributes are retrieved; they might specified from the `[Attributes\]` section referenced in the relevant search.

The `[Response]` section of the `ldapauth.aut` file should list each attribute contained in the `profiledata` attribute.

The `[Response]` section should be configured as follows for `Std_dial` to operate:

```
[Response]  
service-type=  
framed-protocol=  
framed-ip-address=
```

Modifying ldapauth.aut

The following explains how to modify `ldapauth.aut` to support the extensions:

- 1 Add the following field:

```
[Settings]  
UpdateResponse = 1
```


- 2 Add a [GroupedAttributes] section to specify the GlobalProfile and/or ProfileData attributes.

```
[GroupedAttributes]
GlobalProfile = GlobalProfileLDAPAttribute
ProfileData = ProfileDataLDAPAttribute
```

- 3 In the appropriate [Attributes/*name*] section, add the actual LDAP attributes as specified above.

```
[Attributes/name]
GlobalProfileLDAPAttribute
ProfileDataLDAPAttribute
any other attributes
```

- 4 In the [Response] section, set %Profile to the GlobalProfile and list any attributes that are contained in the ProfileData attribute:

```
[Response]
%Profile = GlobalProfileLDAPAttribute
radiusattribute1=
radiusattribute2=
```

Ericsson's e-h235 Authentication Protocol

Steel-Belted Radius now supports Ericsson's implementation of the h-235 Authentication Protocol.

Warning: This protocol is not a complete or strict implementation of the standard and should be used only with Ericsson equipment that supports this feature.

Operation

The user has a password, which is also known to the server. The user sends a name, timestamp, and 20-byte hash to the AAA server. The server validates this digest as follows:

- 1 A 20-byte key is computed from the password. If the password is less than 20 bytes, the key is set to the password with null padding. If the password is greater than 20 bytes, the key is set to the first 20 bytes of the password XORed with the next 20 bytes and so on in wrap-around fashion, until you come to the end of the password.
- 2 The 20-byte hash sent by the user is computed via HMAC-SHA1, where the key is the 20-byte key described above, and the input is the username concatenated with the low-order byte of the timestamp.

This authentication scheme is operative if User-Password (PAP) is active and if the TimeStamp field has been set (see below).

Configuration

The protocol is configured by specifying the attribute that carries the timestamp in the new [e-h235] section of radius.ini:

```
[e-h235]
TimeStamp = Integer-Attribute
```

Uniport Plug-In

3Com's Uniport project can operate with Steel-Belted Radius via a plug-in.

Operation

Uniport requires RADIUS call-type determination as a back-up for SIP call-type determination. To determine call-type, the HiPerARC system sends the Steel-Belted Radius server a request containing a Service-Type attribute of Call-Check (10) and a User-Name attribute in which the value is the same as the Called-Station-Id (DNIS) attribute. The type of call is then determined based on the User-Name (DNIS), and the appropriate Service-Type attribute returned in the Access-Accept packet.

A Uniport plug-in method is instantiated for each value of the Service-Type attribute which can be returned in the Access-Accept. The proper method is utilized using proxy mapping to a directed realm which specifies the method instance. The method then sets the configured profile in the response and indicates it was successful.

The Uniport methods return a reject if a Service-Type attribute with a Call-Check value is not present in the request, if either User-Name or Called-Station-Id attributes are not present, or if their values are not identical.

Configuration

The attribute(s) to be returned in the Access-Accept to identify the call-type are defined as Steel-Belted Radius profiles. For example, a FAX profile may be created which would return a value of 96 in a Service-Type attribute.

The Uniport methods are configured with *.AUT files which specify the profile to return. The method is identified and associated with a directed realm by the initialization string.

For example, a method to return the FAX profile may use a configuration file such as FAX.AUT, which would have the following settings:

```
[Bootstrap]
Enable = 1
InitializationString = UNIPORT FAX
Profile = FAX
LibraryName = Uniport.so
```

The corresponding directed realm would then identify the method in its *.DIR file. For example, in the FAX.DIR file the settings might be:

```
[Auth]
Enable = 1
```

```
[AuthMethods]
Uniport fax
```

The directed realms which refer to the Uniport methods are mapped in the [AuthAttributeMap] of PROXY.INI. A sample map might appear as:

```
[AuthAttributeMap]
Fax
    Service-Type = 10
    Called-Station-Id = 6175471047
FoIP
    Service-Type = 10
    Called-Station-Id = 617*
VoIP
    Service-Type = 10
```

In the example above, the number 6175471047 would be directed to the `Fax` realm, which in turn would use the Uniport method which returns the `Fax` profile. Similarly, 6174976339 would result in a FoIP type and 5085551234 in a VoIP type (as the default).

Important: Wildcards must be listed after any numbers that they might “contain.” In the example above, for example, the 617 wildcard must appear after the number 6175471047, as this last number would be contained within the range of numbers described by the wildcard.*

If you want a default profile, the map may be configured to direct the request to a Uniport method by default. The same result may be obtained by omitting the default from the map and setting the first method in the authentication method chain to the desired default.

If you do not want a default profile, configure the map to direct requests by default to a method which has no profile set; the method returns with a value to indicate a failure to authenticate.

Index

Symbols

%Alias 376, 414
%AllowedAccessHours 369, 420
%AuthType 392
%DN 417
%EffectiveRealm 369, 420
%EffectiveUser 369, 420
%FullName 377, 392, 415
%LoginLimit 375, 413
%Name 369, 420
%NASAddress 369, 392, 420
%NASModel 369, 392, 420
%NASName 369, 392, 420
%OriginalUserName 368, 420
%Password 369, 376, 413, 420
%password 369
%Profile 376, 413
%ProxyRealm 376, 414, 461
%ProxyUserName 376, 414, 461
%Realm 369, 420
%Time 392
%TransactionTime 392
%Type 392
%User 368, 420
%UserName 368, 420

A

Accept 215, 374, 428
Access 185
Accounting 460
Acct section
 dir files 295
 pro files 206
AcctAttributeMap section
 proxy.ini file 269
Acct-Authentic 151
AcctAutoStopEnable 212
Acct-Delay-Time 51, 151
AcctMethods section
 dir files 296
Acct-Status-Type 151, 183, 276, 397, 399
Acct-Termination-Cause 152
ACE/Server 24, 39, 107
activation target number 378
ADD 203
AddPrefix section
 pro files 290, 297
Address Leaks 77
AddSuffix section

 pro files 290, 297
admin directory 8
administrative rights 86
Alias 375, 428
Alias/name section
 account.ini file 178, 186
ALLOW 203
allowed access hours 99
AllowExpiredPasswordsForGroups 219
AllowExpiredPasswordsForUsers 219
Allow-Unmasked-Password 212
Allow-Unmasked-Secret 213
Apply-Login-Limits 213
Attempts 195
AttemptTimeout 195
Attribute mapping 269
attribute mapping 38
Attribute Value Pooling 80
AttributeEdit 213, 268
Attributes 417
attributes 32
Attributes section
 account.ini file 179, 187
Attributes/name section
 aut file 415
Auth section
 dir files 293
 pro files 206
AuthAttributeMap section
 proxy.ini file 269
AuthenticateOnly 213
Authenticate-Only requests 38
Authentication 384, 460
Authentication Methods list 36, 38, 137, 366
AuthMethods section
 dir files 294
Automatic EAP helper 301
Automatic EAP helpers 301
AutoPasswords 213
auto-restart 249
AutoStop 71
AutoStop section
 pro files 284

B

Base 417
Bind 422, 426
Bind Authentication 406
BindName 423, 426

- BindName Authentication 406
- BindPassword 423
- blacklisting 43
- Bootstrap section
 - acc file 396
 - aut file 373, 429
 - sidalt.aut file 222
- BufferSize 181, 189

C

- CachePasscodes 222
- CacheTimeoutAttr 223
- Called-Station-ID section
 - dir files 297
 - pro files 285
- Carryover 181
- CaseSensitiveUsernameCompare 216
- Certificates 423
- Chaddr-prefix 195
- challenge 46
- ChallengeTokenInPassword 223
- CHAP 44
- check-list attributes 54
- CheckMessageAuthenticator 213
- Cipher_Suites 310
- Class attribute 216
- ClassAttributeStyle 214
- Concurrent Tunnel Connections 80
- Concurrent Users 79, 99
- ConcurrentTimeout 380, 396, 399
- Configuration 185
- Configuration section
 - account.ini file 180, 188
 - proxy.ini file 272
 - radius.ini file 212
- Connect 381, 396
- ConnectTimeout 381, 397, 423
- CurrentSessions section
 - radius.ini file 216
- CurrentUsers 185

D

- DefaultResults 378, 381
- Defaults section
 - aut file 421
- DHCP 124, 194
- dictionaries 53
- Dictionary 235
- digest 46
- Directed accounting 292
- directed accounting 51
- Directed authentication 292
- directed authentication 37
- directed realm 292
- DirectedAcctMethods section
 - proxy.ini file 273

- Directory 291
- Discard-After 236
- Discard-Before 236
- DNIS 64
- Domains 102

E

- EAP Identity Response 300
- EAP-Message attribute 300, 302
- EAP-Only 306
- EAP-Type 307
- echo property 56
- Enable 181, 189, 191, 192, 195, 207, 223, 279, 282, 290, 294, 296, 373, 396, 430
- Enc-md5 372, 411
- EventDilutions section
 - events.ini file 200
- EXCLUDE 203
- ExtendedProxy 214, 268
- Extensible Authentication Protocol 300
- external accounting 49
- external authentication 37

F

- FailedAuthOriginStats section
 - radius.ini file 216
- Failover 125, 126
- Failure section
 - aut file 374, 428
- fast-fail 69
- FastFail section
 - pro files 70, 289
- FileSystemFreeKBWarningClear 202
- FileSystemFreeKBWarningIssue 202
- Filter 418
- FilterIn 279, 282
- FilterOut 279, 282
- First-Handle-Via-Auto-EAP 307
- FlashReconnect 423, 426
- Framed-Compression 55
- FramedIPAddressHint 76, 214
- FullName 375, 428

G

- GroupedAttributes 469
- Groups section
 - access.ini file 177

H

- Hint 76
- Hlent 195
- Host 423
- htype 195
- HUP signal 232

I

Ignore-Acct-Ss 235
Ignore-Ports 236
ImportExport 185
In 460
IncludeProxy 191
InitializationString 40, 223, 366, 373, 430
install.sh 6
IP Address Pool field 89
IP Address Pools 121, 216, 355
IP-Pools 185
IPPoolSuffixes section
 radius.ini file 217
IPX Address Pools 128, 355
IPX-Pools 185

J

Java 12

L

LastResort 423
LDAP section
 radius.ini file 218
LDAPAddresses section
 radius.ini file 218
LdapVersion 423, 426
LeaseTime 197
LibraryName 373, 396, 430
License 185
LineSize 181, 189
load balancing 401
LocalPort 196
Lockout 207
lockout 41
log files 49
LogAccept 145, 215
LogDir 214
LogLevel 145, 214, 381, 426
LogReject 145, 215

M

Macro records 245
Make/model field 53, 88
Management Information Base 324
mapping section
 servtype.ini file 455
MaxConcurrent 370, 381, 394, 397, 399, 424
Max-EAP-Fragment 236
MaxPong 192
MaxShutdown 193

MaxSize 181, 189
MaxStartup 193
MaxWaitReconnect 381, 397, 424
MD4 372, 411
MessageAuthenticator 279
MessageID 223
MIB 324
MinFailures 289
MinLeaseTime 197
MinSeconds 289
ModifyUser section
 dir files 297
 pro files 290
MS-CHAP 44
MS-CHAP-V2 45
Multiple Servers 127
multi-valued attributes 55

N

Native User 36, 96
Native User authentication 36
NTDomain section
 radius.ini file 218
NumAttempts 280, 282

O

OnFound 417, 427
OnNotFound 417, 427
Oracle 4, 9, 367, 373, 379, 385, 390, 395, 396, 397
orderable attributes 56
Out 460
OverallTimeout 195

P

Pad 196
PAP 44
ParameterMarker 381, 397
pass-through authentication 37
Password 424, 427
password output parameter 370
PasswordCase 427
PasswordFormat 382, 427
PEAP_Max_Version 310
PEAP_Min_Version 310
perfmom counters 163
perfmom.exe 163
Phantom records 80
PhantomTimeout 215
PingInterval 193
PoolPctAddressAvailWarningClear 202

- PoolPctAddressAvailWarningIssue 202
- Pools section
 - dhcp.ini file 196
- Port 424
- Ports section
 - radius.ini file 35, 220
- Prefetch-capable 303
- PrequalifyChecklist 219
- PrivateDir 215
- Processing section
 - proxy.ini file 275
- Product-Scan-Acct 236
- Product-Scan-Auth 236
- Profile 191, 374, 428
- profile 96
- ProfileForExpiredUsers 220
- ProfileForExpiredUsersInGroups 220
- Profiles 185
- profiles 58
- Proxy 185, 460
- proxy 30
- Proxy Accounting 116
- Proxy Authentication 116
- Proxy AutoStop 71
- proxy RADIUS 37, 49
- proxy RADIUS accounting 49
- proxy RADIUS authentication 37
- Proxy RADIUS realms 66
- ProxyFastFail 70, 215, 289
- ProxySource 215
- ProxyStripRealm 215

Q

- QueryTimeout 382, 397, 399, 424
- QuoteBinary 182, 189
- QuoteInteger 182, 190
- QuoteIPAddress 182, 190
- QuoteText 182, 190
- QuoteTime 182, 190

R

- radius.dct 53
- radiusclass 339
- RAS-Clients 185
- realms
 - Proxy RADIUS 66
- Realms section
 - proxy.ini file 275
- RecordLocally 282, 296
- RegularExpression 227
- Reject 215
- Rejection Messages 135
- Rejects 207
- REPLACE 203
- Report 185
- REPORT.RTF 161

- Request section
 - aut file 419
 - pool files 197
- RequestTimeout 280, 282
- RequestTimeoutMills 280, 283
- ReserveMemoryKB 202
- ResetSeconds 289
- Resource Domain 17
- Response section
 - aut file 412
- Results section
 - aut file 375
- retry policy 69, 115
- RetryInterval 291
- Return-List attributes 55
- Rollover 182, 190
- RolloverOnStartup 182, 190
- RolloverSeconds 290
- RolloverSize 290
- RoundRobin 280, 283, 287, 288

S

- Scope 418
- Search 417, 424, 427
- Search/name section
 - aut file 416
- SecondsToCachePasscodes 222
- SecurID 24, 107
- SecurID section
 - radius.ini file 221
- Self section
 - radius.ini file 224
- Send-Class-Attribute 235
- Send-Session-Timeout-on-Challenge 235
- server directory 6
- Server section
 - aut file 378, 424
- Server/name section
 - aut file 379, 421
- ServerInfo section
 - tacplus.ini file 231
- ServerPort 195
- Settings section
 - account.ini file 181, 189
 - aut file 380, 425
 - blacklist.ini file 191
 - bounce.ini file 192
 - dhc files 197
 - dhcp.ini file 195
 - lockout.ini file 207
 - servtype.ini file 454
 - sidalt.aut file 222
- Shared secret 89
- shared secret 34
- SharedSecret 231
- ShutdownDelay 52, 291

- sidalt.aut 222
- Smart Static Accounting 277
- SpooledAccounting section
 - pro files 290
- SQL 382, 399
- SSL 424
- Static proxy accounting 276
- StaticAcct section
 - proxy.ini file 276
- StaticAcctProxy section
 - radius.ini file 224
- StaticAcctRealms 283, 296
- Statistics 185
- Strip section
 - aut file 224, 383
- StripRealm 280, 283, 294, 296
- SuccessResult 382
- Suppress section
 - events.ini file 201
- system assigned values 56

T

- TACACS+ 26, 39, 46, 109, 231
- TargetAddress 197
- TargetHost 231
- TargetsSection 279, 280, 281, 283
- ThreadAvailWarningClear 202
- ThreadAvailWarningIssue 201
- Thresholds section
 - events.ini file 201
- Timeout 427
- Titles 182, 190
- TokenAttr 223
- TraceLevel 145, 216
- TreatAddressPoolsAsDisjoint 216
- Tunnel name parsing 136
- Tunnels 60, 118, 185, 354
- Type section
 - acc file 397
- Type/statement section
 - acc file 399
- TypeNames section
 - acc file 399
 - account.ini file 183

U

- UDPAcctPort 35, 220
- UDPAuthPort 35, 220
- UNIXcrypt 372, 410, 411
- update.ini 232

- UpperCaseName 382, 397, 427
- UseNewAttributeMerge 216
- User type field 54
- User-Name 227
- User-Name attribute 300
- UserNameTransform section
 - radius.ini file 460
- Users 186
- Users section
 - access.ini file 177
- USR2 signal 232
- UTC 182, 190, 397, 427

V

- ValidateAcct section
 - radius.ini file 227