

STEEL-BELTED RADIUS™

Administration Guide ***Windows NT Version***

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

617 497-6339
617 491-6503 (technical support)

© Copyright 1996-1998 Funk Software, Inc. All rights reserved.
5th edition, May 1998.

Steel-Belted Radius © 1996-1998 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, 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.

Table of Contents

Introducing Steel-Belted Radius	5
Overview	6
What is RADIUS?.....	6
What is Steel-Belted Radius for Windows NT?	7
System Requirements.....	8
Steel-Belted Radius Licensing.....	8
Documentation.....	8
Technical Support	9
 Installing Steel-Belted Radius	 11
Installing Steel-Belted Radius for Windows NT	12
Starting and Stopping the RADIUS Service	14
Upgrading from a 30-Day Trial Installation.....	14
What's Next After Installation?	14
 RADIUS Concepts	 17
The RADIUS-Based Remote Access Environment	18
RADIUS Authentication	19
RADIUS Accounting	27
Proxy RADIUS.....	27
External Databases	31
Tunnels	32
Resource Management	37
 Administering Steel-Belted Radius	 43
The Steel-Belted Radius Administrator	44
The Servers Dialog.....	45
The RAS Clients Dialog.....	46
The Users Dialog	52
The Profiles Dialog	70
The Proxy Dialog.....	73

The Tunnels Dialog	78
The IP Pools Dialog	80
The IPX Pools Dialog	84
The Access Dialog	87
The Configuration Dialog.....	89
The Statistics Dialog	93
Reporting	101
Import/Export.....	103
Logging and Monitoring Features	107
Logging Features.....	108
Performance Monitor Graphing	108
Authentication Log File.....	108
Accounting Log File.....	109
Initialization and Dictionary Files	115
Initialization Files	116
Dictionary Files.....	121
Using Steel-Belted Radius with SQL	129
Using Steel-Belted Radius with SQL.....	130
Authenticating to an SQL Database	130
SQL Accounting	139

Introducing Steel-Belted Radius

1

Overview

Thank you for selecting Steel-Belted Radius as your remote access authentication solution.

With Steel-Belted Radius for Windows NT, you'll be able to administer a single service that provides for centralized authentication of all the users that connect to your LAN via remote-access servers, firewalls, or other means.

Plus, Steel-Belted Radius is completely integrated with Microsoft networking. So all of the authentication information that you've already set up in NT domains and on individual NT hosts can immediately be used to provide dial-in security as well.

What is RADIUS?

RADIUS is a standardized method of information exchange between a device that provides network access to users (such as a Remote Access Server used for network dial-up) and a device that contains authentication and profile information for these users (such as Steel-Belted Radius).

RADIUS stands for "Remote Authentication Dial In User Service." The RADIUS standard was initially developed by Livingston Enterprises and has recently become a standard of the Internet Engineering Task Force (IETF).

The RADIUS standard is quickly becoming the preferred way to perform authentication for dial-up users. With RADIUS, you can:

- ♦ Centralize administration of user information across all your Remote Access Servers performing dial-in and firewall authentication.
- ♦ Utilize security information to which your Remote Access Servers would otherwise have no access (such as the security databases of Windows NT domains and individual NT hosts).
- ♦ Deploy Remote Access Servers from a variety of vendors while maintaining a common security model and administrative interface.

What is Steel-Belted Radius for Windows NT?

Steel-Belted Radius for Windows NT is a complete implementation of the RADIUS standard that runs as a Windows NT service.

Because Steel-Belted Radius is tightly integrated with Windows NT, you can use the passwords and groupings you've already created in Windows NT Domains or on Hosts as the basis for authenticating remote users dialing in to any of your Remote Access Servers.

Centralize Administration of All Your RAS's

Using Steel-Belted Radius, you'll be able to enter user profiles that determine which of your users are authorized to dial in or connect via a firewall to the network and what type of connection each user is permitted to make. With Steel-Belted Radius, it is no longer necessary to separately set up user profiles on each Remote Access Server; each RAS relies on Steel-Belted Radius as an authentication server to determine the rights of each user from a single, common database that you can administer easily.

Leverage User Information Already in Windows NT

Not only won't you have to separately administer RAS's, you also won't have to separately administer each user. Because Steel-Belted Radius is tightly woven into the Microsoft networking fabric, you'll be able to specify profiles for many users all at once using group names you've already defined. And Steel-Belted Radius works with the security you've already established: each user can be authenticated against NT Domain security or any individual NT Host using his or her current password. And you can combine Domain, Host, and the Native authentication database to provide multiple sources of authentication information.

Use Steel-Belted Radius to Service RAS's from Different Vendors

Because RADIUS is a standard, Steel-Belted Radius can be used with any RAS device that implements that standard. However, it is also possible for individual dial-in or firewall RAS vendors to create proprietary extensions to serve the particular needs of their servers.

Steel-Belted Radius has already incorporated proprietary extensions from a number of vendors. Steel-Belted Radius has a powerful and flexible technique that allows it to accommodate specific extensions from any vendor. All

informational tokens that are passed between a RAS and Steel-Belted Radius are described in a set of specially formatted ASCII files called dictionaries. There is one basic dictionary that describes all standard tokens common to all implementations, and for each RAS vendor a separate dictionary describes the specific extensions of that vendor.

System Requirements

Steel-Belted Radius for Windows NT runs as a service on any Windows NT workstation or server (version 4.0 or later), with TCP/IP properly configured.

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

Steel-Belted Radius supports several SQL databases for use as external databases for RADIUS authentication and accounting, including:

- ♦ Oracle WorkGroup Server v7.3.3, v7.3.4
- ♦ Informix Dynamic Server version 7.2x

Refer to the **readme.txt** file for a comprehensive list.

Steel-Belted Radius Licensing

Steel-Belted Radius for Windows NT may be installed on a single Windows NT workstation or server.

For more information on licensing, please refer to the enclosed license agreement or contact Funk Software, Inc. directly.

Documentation

This manual describes how to install and administer Steel-Belted Radius for Windows NT.

Most of the information in this manual is also contained in the on-line help available from the Administrator program.

An additional help file, available from the **Vendor info** menu item under **Help**, contains useful information about a variety of Remote Access Servers, Firewalls, and other equipment that can operate as a RADIUS client. Be sure to consult this help file when configuring your equipment for use with Steel-Belted Radius.

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 at no charge.

- ♦ This manual and the **readme.txt** files on your diskettes 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.
- ♦ You can send e-mail to our Internet address **support@funk.com**.
- ♦ Contact our world wide web server at **http://www.funk.com**.
- ♦ We provide 30 days of technical support by phone at no charge, starting from your first support call. Our technical support staff is available to assist you at **(617) 491-6503**, on weekdays between 9:00 AM and 5:30 PM Eastern Standard Time.

For support beyond the initial 30-day period, we offer a range of support options including support and maintenance contracts and pay-per-call. Consult the enclosed "Technical Support and Service Offerings" brochure for the support plan that best meets your needs.

If you are located outside North America, please refer to the enclosed list of Authorized International Partners for the name of the support provider in your country.

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

Installing Steel-Belted Radius

2

Installing Steel-Belted Radius for Windows NT

This chapter describes how to install the Steel-Belted Radius service onto a Windows NT server or workstation.

You first need to pick an appropriate NT machine on which to install Steel-Belted Radius. The NT machine's role in your Microsoft network has implications as to the types of authentication it will be able to perform.

Before installing, please read the "RADIUS Authentication" section of Chapter 3, "RADIUS Concepts." Pay careful attention to the subsection entitled "Domain vs. Host Authentication." This subsection describes the two types of authentication that rely on Microsoft networking. Based on the type(s) of authentication you'll want to do, you'll be able to pick an appropriate NT machine to install the service on.

To install Steel-Belted Radius for Windows NT:

- 1 Run SETUP.EXE from the Steel-Belted Radius installation disk.
Insert the disk in drive **A:**, click **Start**, select **Run**, then enter **A:\SETUP ↵**.
- 2 The Software License Agreement screen appears. Before proceeding, make sure that you read and agree with the terms of the license agreement.
Click **Yes** if you agree. Otherwise, click **No**.
- 3 The Welcome screen appears.
Click **Next** to continue.
- 4 The Select Components screen appears. For a normal installation, make sure both **RADIUS Admin Program** and **RADIUS Server** are checked. You may use the default destination directory for each component you are installing, or click **Browse** to select a different directory.
Click **Next** to continue.
- 5 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.
- 6 The License Key screen appears.
Enter the **License key** printed on your Steel-Belted Radius license agreement card, or check the **Install 30-day trial** box, as appropriate.
Click **Next** to continue.

- 7 The Service Login Account screen appears. Enter the **User Name** of the account under which you wish to run the Steel-Belted Radius service, and enter the **Password** (twice).

NOTE: This screen will not appear if you are installing onto an NT server which is a Domain Controller. In that case, the service will run under the system account, rather than under a user account, and Host authentication will not be available.

If the account already exists, then it will be used with the password you entered. If you enter a new **User Name**, it will be created for you automatically with the password you entered.

Click **Next** to continue.

- 8 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.

- 9 Once installation is completed, the Setup Complete screen will appear. This screen gives you the opportunity to view the **readme.txt** file and start the Steel-Belted Radius Administrator.

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

Updating a Previous Installation

NOTE: If you have previously installed a copy of Steel-Belted Radius for Windows NT, you must be sure that it is not running on the server when installing the new copy.

If you have previously installed a copy of Steel-Belted Radius for Windows NT, you will receive a warning that files may be overwritten with newer versions. If you receive this message, you should do the following:

- 1 Export all data from the previous installation.
NOTE: For export/import instructions, see Chapter 4, "Administering Steel-Belted Radius."
- 2 Back up the Steel-Belted Radius server directory (usually **\winnt\system32**).
- 3 Complete the installation procedure as described above.
- 4 Import the old data into the new Steel-Belted Radius installation.

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.

If you don't want it 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 it runs.

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 to your existing installation:

- 1 Purchase the Steel-Belted Radius software, either by contacting your preferred reseller or by contacting Funk Software, Inc. directly.
You will be shipped a product package that will contain a license key. This key will convert your 30-day trial software to an unlimited version.
- 2 Start the Steel-Belted Radius Administrator.
- 3 Select **File License**.
The Add a License for Server dialog displays.
- 4 Enter the license key and click **OK**.
The next time Steel-Belted Radius is started, the license will be loaded.

What's Next After Installation?

Once you've installed Steel-Belted Radius, you still need to configure it before it becomes usable. The steps, in general, are as follows:

- 1 Configure each of your RAS's to act as a RADIUS client.
Each RAS must be configured with the IP address of the Steel-Belted Radius server, a secret (password) that is shared with the server, and the make/model of the Remote Access Server.

- 2 Make sure the machine on which you're running Steel-Belted Radius has the IP protocol configured
- 3 Run the Steel-Belted Radius Administrator program.
- 4 From the Servers dialog, connect to your Steel-Belted Radius server.
- 5 From the RAS Clients dialog, provide information about each of the Remote Access Servers configured to act as a RADIUS client.

Configuration information includes the IP address of the Remote Access Server, the shared secret, and the make/model information of the Remote Access Server. If a specific make/model is not listed, use **Standard Radius**.
- 6 From the Users dialog, identify each of the users or groups of users that are permitted to dial in to the Remote Access Servers. Set up user attributes, either by assigning them in the Users dialog or by creating user profiles in the Profiles dialog.
- 7 For SecurID to work with the Steel-Belted Radius server, you must place the **sdconf.rec** file in the **\winnt\system32** directory. If you place the file in this directory after Steel-Belted Radius has been started, then you must stop and start Steel-Belted Radius before SecurID will work.

It is also important that you configure the Steel-Belted Radius server as a client of the ACE/Server. From the ACE/Server **Client** menu, choose **Add Client**. Complete the Client dialog, giving the Steel-Belted Radius server a **Client type** of **Net OS Client**.
- 8 For pass-through authentication to a TACACS+ server to work, the **tacplus.ini** file must be present in the **\winnt\system32** directory. This happens automatically following Steel-Belted Radius installation.

You must edit **tacplus.ini** to identify the shared secret and host machine that you use for TACACS+ (for details, see Chapter 6 "Initialization and Dictionary Files"). If you edit **tacplus.ini** after Steel-Belted Radius has been started, then you must stop and restart Steel-Belted Radius before your changes will take effect.

The remainder of this manual provides detailed information about using the Administrator program to configure Steel-Belted Radius.

You may consult our online "RAS Product 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 **RAS Clients**, and clicking the **Vendor Info** button on the RAS Clients

dialog. You can also access the file by selecting **Help Vendor Info** from the Administrator menu bar.

For more detailed information about configuring your RAS's and Firewalls, consult the manufacturer's documentation provided with each unit.

RADIUS Concepts

3

The RADIUS-Based Remote Access Environment

The RADIUS-based remote access environment has three components: Users, Remote Access Servers, and the RADIUS server. Each user is a client of a RAS; each RAS is both server to the user and client of the RADIUS server.

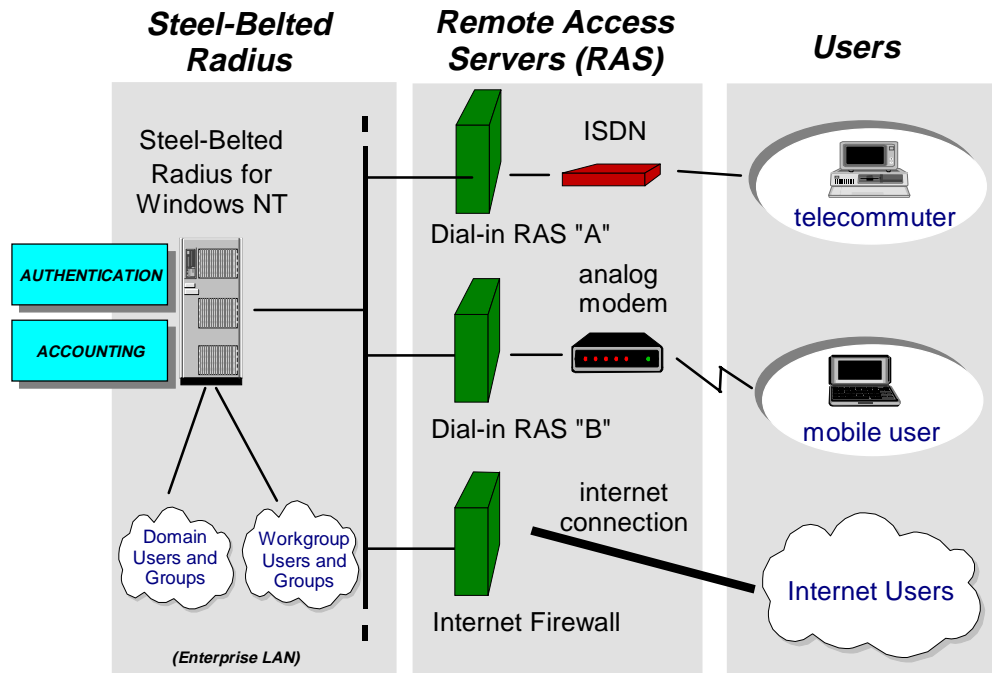


Figure 3-1: Steel-Belted Radius Elements

User

The user is the person trying to gain access to the network from home or from the road.

Typically, the user has a SLIP or PPP dialer that allows him or her to dial into a Remote Access Server at the enterprise LAN and become a remote node on the enterprise network, with IP and/or IPX access to network resources.

Remote Access Server

The Remote Access Server (or RAS) is a device that:

- ◆ Supports dial-in such as SLIP or PPP dial-in calls, authenticates each user via the RADIUS Server, and then routes that user onto the network.
- ◆ Supports direct connections to the network through a firewall, authenticates each user via the RADIUS Server, and then grants network access with specific capabilities.

Examples of dial-in RAS's which support the RADIUS standard include the Ascend MAX, Bay Networks' Annex, Shiva LAN Rover, ITK NetBlazer, and US Robotics' Total Control; firewall products include Check Point's FireWall-1 and Raptor's Eagle.

Most RAS devices can handle multiple dial-in users at once, and the corporate network might include a single RAS or multiple RAS's working in tandem.

RADIUS Server

The RADIUS Server accepts authentication requests from one or more Remote Access Servers, performs the authentication, and responds with the result: either an accept or a reject. The RADIUS server also provides Accounting services, if the RAS can support this.

A typical installation will include a single RADIUS server, for example, Steel-Belted Radius, to handle all the Remote Access Servers. Companies with Remote Access Servers at multiple sites could elect to have a separate RADIUS Server at each site; or, if the various sites were linked over a WAN of reasonable speed or over the Internet, a single RADIUS server could be made to handle multiple Remote Access Servers at multiple sites.

RADIUS Authentication

The primary function of the RADIUS server is authentication. This section covers the following areas critical to an understanding of authentication:

- ◆ What happens during the authentication process
- ◆ Types of authentication available
- ◆ RADIUS attribute exchange
- ◆ Dictionary files

What Happens During Authentication

It may be instructive to follow the steps involved in a typical transaction in which a user attempts to gain access to the network via RADIUS authentication.

- 1 A user dials in to one of several RAS's and PPP negotiation begins.
- 2 The RAS passes authentication information developed during PPP negotiation to the RADIUS server.
- 3 If the RADIUS server is able to authenticate the user, it issues an **accept** response to the RAS, along with profile information required by the RAS to set up the connection (this might include IP address, maximum connect time, and the like).

If the RADIUS server is unable to authenticate the user, it issues a **reject** response to the RAS, along with a text string indicating the reason.
- 4 Using this information, the RAS completes PPP negotiation with the user:

If the RAS received an **accept** response, it can now allow the user to begin operating on the network.

If the RAS received a **reject** response, it terminates the user's connection, possibly passing on the reason for termination for display at the user terminal.

Authentication Types

During an authentication transaction, password information is transmitted between the RAS and the RADIUS server. The password information is encrypted using a secret key that you enter both at the RAS and at the RADIUS server.

The password information originally comes from the user, usually as part of PPP negotiations. The RAS is really just an intermediary here, and it is best to think of authentication as being a transaction between the user and the RADIUS server.

There are three types of authentication transactions used between a remote access user and RAS. Each represents a method of authentication used in PPP:

- ♦ **PAP** (Password Authentication Protocol) is very simple. The user sends his or her password to the RADIUS server, and the RADIUS server validates it, either against its own database or against the security database of a Windows NT domain or an individual NT host.

Of the two legs of the journey the password takes between user and RADIUS server, the first leg is usually unencrypted, and the RAS gets the password from the user in clear text. For the second leg, the RAS encrypts the password and the RADIUS server decrypts it using a shared secret key.

Ultimately, the RADIUS server has the password in clear text form and is able to make use of it directly for authentication.

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

With CHAP, the RAS generates a random number (the challenge) and sends it to the user. The user's PPP client creates a "digest" (a one-way encryption) of the password concatenated with the challenge, and sends this digest to the RAS. Because the digest is a one-way encryption, the RADIUS server cannot recover the password from the digest. What it can do is perform the identical digest operation using its own copy of the user's password stored in its database; if the two digests match, the user is authenticated.

- ♦ **MS-CHAP** (Microsoft Challenge Handshake Authentication Protocol) is a Microsoft authentication protocol which, like CHAP, avoids sending passwords in clear text. MS-CHAP is tightly integrated into Microsoft NT security.

Steel-Belted Radius supports the MS-CHAP protocol for the following authentication methods only: **Native User, Domain User, Domain Group, Host User, Host Group**, and **External Database** (if the password is stored in clear text form only).

Steel-Belted Radius Authentication Methods

Steel-Belted Radius has a number of methods for performing the actual authentication. These methods are as follows:

- ♦ Authentication against a local database
- ♦ Pass-through authentication against Microsoft networking
- ♦ Pass-through authentication against token-based systems
- ♦ Authentication against a proxy database
- ♦ Authentication against an external database

Only **Native User** authentication against a local database permits the use of either PAP, CHAP, or MS-CHAP protocols. Pass-through to Microsoft

Domain User or **Domain Group** authentication permits the use of PAP or MS-CHAP. Pass-through to **TACACS+** permits the use of PAP or CHAP. The other authentication techniques require the use of PAP exclusively. Use of an **External Database** permits the use of CHAP or MS-CHAP, but only if the password is stored in clear text form.

Authentication Against a Local Database

- ♦ **Native User**

The Steel-Belted Radius server authenticates the user directly against its own local database.

For each Native user you must create a separate entry using the Administration program, giving the user's name, password and other information.

Pass-through Authentication Against Microsoft Networking

- ♦ **Domain User**

Each Domain User entry specifies the name of a Domain and the name of a User in that Domain.

In order for a dial-in user to authenticate against a Domain User entry, the user's name must match that of the Domain User entry, and the password supplied must be capable of logging the user into the specified Domain.

- ♦ **Domain Group**

Each Domain Group entry specifies the name of a Domain and the name of a Group in that Domain.

In order for a dial-in user to authenticate against a Domain Group, the user must be a member of the specified Group, and the password must be capable of logging the user into the specified Domain.

- ♦ **Host User**

Each Host User entry specifies the name of an individual Windows NT Host, and the name of a User defined on that Host.

In order for a dial-in user to authenticate against a Host User entry, the user's name must match that of the Host User entry, and the password must be capable of logging the user into the specified Host.

♦ **Host Group**

Each Host Group entry specifies the name of an individual Windows NT Host, and the name of a Group defined on that Host.

In order for a dial-in user to authenticate against a Host Group entry, the user must be a member of the specified Group, and the password must be capable of logging the user into the specified Host.

The types of authentication that rely on Microsoft networking use either of two techniques: Domain authentication or Host authentication. It is important to understand the implications and limitations of each of these techniques.

Most medium to large Microsoft networking installations are organized into one or more Domains for security purposes. If your network is so organized, then Domain authentication should be preferred over Host authentication for the purposes of RADIUS security.

Host authentication is provided to accommodate networks that aren't organized into Domains, or networks that may have mixed Domain and Workgroup security.

Note that the two techniques are not mutually exclusive.

♦ **Domain**

Domain authentication lets you specify users and groups that you've defined within one or more Domains. Each Domain authentication entry consists of the name of a Domain plus the name of a User or Group defined in that Domain.

In order to use Domain authentication, the Steel-Belted Radius service must be running on an NT machine 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 "trusts" the Domain of the Steel-Belted Radius service. The inverse trust relationship is immaterial; that is, it only matters who trusts the RADIUS Domain, not who the RADIUS Domain trusts.

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

- ◆ **Host**

With Host authentication, you specify users and groups that you've defined on an individual Windows NT Host. Each Host authentication entry consists of the name of an NT Host plus the name of a User or Group defined on that Host. It doesn't matter whether the NT Host is a workstation or a server. Steel-Belted Radius performs a peer-to-peer login and logoff to authenticate a user on a particular Host.

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 machine on which you install Steel-Belted Radius must not be a Domain Controller.

Due to a limitation in Microsoft networking, peer-to-peer authentication will not work from a Domain Controller.

- ◆ The NT machine 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.

The reason for this is simple: There can be only one peer-to-peer connection between two machines at any given time.

Steel-Belted Radius performs Host authentication by attempting to log the user into the specified Host. But if the RADIUS machine is already logged into that Host (under that name or any other name), the new login attempt will fail due to the one-connection restriction.

Pass-through Authentication Against Token-based Systems

- ◆ **SecurID**

Each SecurID entry specifies the name of a user to be authenticated via SecurID. The user must enter a PIN and token code as a password; this information will be passed on to an ACE/Server for authentication.

*NOTE: For SecurID to work, you must place the **sdconf.rec** file in the **lwinnt\system32** directory. If you place the file in this directory after Steel-Belted Radius has been started, then you must stop and start Steel-Belted Radius before SecurID will work. You must also add the Steel-Belted Radius to the ACE/Server's list of clients, with a **Client type** of **Net OS Client**.*

Pass-through Authentication Against TACACS+

- ◆ **TACACS+**

Each TACACS+ entry specifies the name of a user to be authenticated via TACACS+. The user must provide a username and password; this information will be passed on to a TACACS+ Server for authentication.

*NOTE: For pass-through authentication to a TACACS+ server to work, the **tacplus.ini** file must be present in the **lwinnt\system32** directory. This happens automatically following Steel-Belted Radius installation.*

*You must edit **tacplus.ini** to identify the shared secret and host machine that you use for TACACS+ (for details, see Chapter 6 “Initialization and Dictionary Files”). If you edit **tacplus.ini** after Steel-Belted Radius has been started, then you must stop and restart Steel-Belted Radius before your changes will take effect.*

Authentication Against a Proxy Database

- ◆ **<Proxy Server>**

Proxy authentication allows the Steel-Belted Radius server to pass the user’s login information on to a proxy server for authentication.

NOTE: For details on how this works, see “Proxy RADIUS” below.

Authentication Against an External Database

- ◆ **<SQL Database>**

Using the username and password supplied in a user’s access request, the Steel-Belted Radius server can select records in an external SQL database to authenticate that user. Several different SQL databases are supported.

NOTE: For details on how external database authentication works, see “External Databases” below.

RADIUS Attribute Exchange

The authentication transaction serves an additional purpose beyond simply authenticating the user.

Along with the authentication information that the RAS includes as part of a RADIUS request, the RAS also passes information about the type of connection

the user is trying to establish. The RADIUS server can use this information to further qualify the user, possibly issuing a reject based on this information.

Similarly, the RADIUS server includes additional information as part of the accept response it issues to the RAS. The RAS uses this information to control various aspects of the user's connection.

This aspect of the authentication transaction is called "attribute exchange."

Attribute exchange is controlled by the user's profile. Each profile lists attributes of two types: check-list attributes and return-list attributes.

Check-list Attributes

Check-list attributes define a set of requirements for the connection. During the authentication transaction, the RAS must send attributes to the RADIUS Server that match the check-list; if they don't, the RADIUS server will issue a reject even if the user can be authenticated.

For example, by including appropriate attributes in the check-list, a variety of rules could be enforced. Only certain users might be permitted to use ISDN connections, or dial in to a particular RAS. Or, Caller ID could be used to validate a user against a list of legal originating phone numbers.

Return-list Attributes

Return-list attributes are the attributes that the RADIUS server sends back to the RAS once authentication is successful. The return-list defines additional parameters that the RAS should assign to the connection, typically as part of PPP negotiations.

For example, 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.

Dictionary Files

The RADIUS server uses Dictionary files to establish check-list and return-list attribute values. The Dictionary file contains the RAS-specific, proprietary items which may be set for a user. For more information, please see the Dictionary file for your particular RAS in the installation directory.

RADIUS Accounting

RADIUS Accounting is an additional feature of the RADIUS standard that permits a RADIUS server to track when users start and stop their dial-in connections and to acquire statistics about each session.

Using RADIUS Accounting, the RADIUS server can maintain:

- ♦ A history of all user dial-in sessions, indicating start time, stop time, and various statistics for the session.
- ♦ A real-time snapshot of all current usage, indicating which users are currently connected to which Remote Access Servers.

Many Remote Access Servers that support RADIUS also support RADIUS Accounting.

Steel-Belted Radius fully supports RADIUS Accounting. All Accounting transactions are logged to a comma-delimited file that can be imported into spreadsheets and database programs for report generation and billing.

Current Users Display

One of the most useful capabilities enabled by RADIUS Accounting support is a real-time list of active RADIUS users. This Current Users display is available from the Steel-Belted Radius Administration program's Statistics dialog. For every active dial-in session, a line is displayed identifying the user, the RAS, the port number, the assigned IP address, and other information.

Proxy RADIUS

Proxy RADIUS refers to the forwarding of a request from one RADIUS server to another.

Normally, when a RADIUS server receives an authentication request from a RAS, it processes it locally and replies directly to the RAS. However, it is also possible for the RADIUS server to act as intermediary, or "proxy," between the RAS and a second RADIUS server, called the "target" server. The proxy server receives the request, forwards it to the target server, waits for a reply, then passes the reply back to the RAS that originated the request.

Proxy RADIUS is normally used in situations where the local RADIUS server may have adequate information to authenticate some users, but must rely on other RADIUS servers for the remainder.

Steel-Belted Radius is fully capable of acting as a proxy as well as acting as the target of a proxy.

Steel-Belted Radius provides the following proxy features:

- ♦ Proxy forwarding to other RADIUS servers based on username parsing
- ♦ “Roaming” proxy forwarding, using DNS (Domain Name System) to locate target RADIUS servers. This powerful feature relieves the administrator from having to pre-configure possible target servers in complex systems
- ♦ Proxy Accounting options for forwarding or retention of accounting transactions, or both
- ♦ Full capability to act as a target server for other proxy servers
- ♦ Proxy forwarding as an “official” or default authentication method

Configuring Proxy Forwarding

To set up proxy forwarding, you must use the Proxy dialog in the Administrator program to add a database entry for each target server, specifying its name, its IP address, and a shared secret.

Steel-Belted Radius will then forward any request to the appropriate target server when the username it receives from the RAS is of the following form:

user_name@target_name

where:

user_name is the name of the dial-in user

target_name is the name of the target RADIUS server

Steel-Belted Radius first looks up **target_name** in its database of target servers, gets its IP address and shared secret, and forwards the request to that IP address encrypted with the shared secret. Note that **@target_name** will be stripped from the username in the forwarded request.

NOTE: The proxy server and the target server must be configured with the same shared secret in order for forwarding to be successful. However, the secret shared between the RAS and the proxy server need not be the same as the secret shared between the proxy server and the target server. The proxy server decrypts

requests received from the RAS using one shared secret, then re-encrypts the request forwarded to the target server with the other shared secret.

Roaming

If you are setting up more than a few target servers, you might want to consider using the “roaming” feature. With roaming, you’ll be able to add new target servers easily, without having to add each new target server to each proxy’s database.

If you add a special entry to the proxy’s target database called **<ROAMING>**, then Steel-Belted Radius will consider any **target_name** that is not in its database as the name of an IP host that it should resolve to an IP address using DNS. Provided the DNS lookup succeeds, it will forward the request to the target server at that IP address. The request will be encrypted using the shared secret specified in the **<ROAMING>** entry.

Note that to use the roaming feature, one single shared secret must be used to configure all target servers.

A Proxy Example

Suppose a major multinational corporation has offices in Aspen and Bayonne. Each office has a Steel-Belted Radius server, which is locally administered to authenticate all locally-based users. Users from either city are welcome to access the network in the other city as well; they will be authenticated by proxy.

To set up such a network, at each Steel-Belted Radius server simply add the other city as a proxy target server. Now user Alice from Aspen can access the Aspen LAN as **Alice**, or the Bayonne LAN as **Alice@Aspen**; the Bayonne server will forward the request to Aspen.

Steel-Belted Radius as a Target Server

Steel-Belted Radius can act as a target server, receiving inbound requests from a proxy server just as it might receive requests from a RAS.

To allow the target server to receive requests from a proxy, you must use the RAS Clients dialog on the target server to add an entry for the proxy server. In other words, you must treat the proxy server as if it were just another RAS client, specifying its IP address and a shared secret.

If you’d like to be able to accept proxy requests from any IP address, you can add a special RAS Client entry called **<ANY>**, and specify a shared secret. This

entry permits forwarded requests from any proxy server to be accepted, provided the shared secret is correct.

Dictionary Handling for Inbound Proxy Requests

Dictionaries are handled somewhat differently for inbound proxy requests. Normally, the **Make/model** field of the RAS client entry for the proxy server will determine which dictionary is used. However, your target server may receive requests via a single proxy server that originated from multiple RAS devices, possibly of different types. The single **Make/model** entry for the proxy server may not be adequate to handle the variety of RAS's on the other side of that server.

To handle this problem, you can add the originating RAS's to your list of RAS Clients on an exception basis in order to specify the appropriate dictionary. Steel-Belted Radius will use the **Make/model** of the originating RAS, if it is present, in preference to the **Make/model** of the proxy server.

Proxy Forwarding as an Authentication Method

You can configure proxy forwarding as an authentication method in the Steel-Belted Radius Administrator. As explained above, when you set up proxy forwarding, you create a proxy entry using the Proxy dialog. The **Authentication Method Status** panel on the Proxy dialog allows you to decide whether or not your proxy entry can be included in the **Authentication Methods** list on the Configuration dialog. If so, this proxy's name appears in the **Authentication Methods** list along with all the other methods of authentication that are available to this Steel-Belted Radius server. The proxy can be assigned a priority in this list just like all the other methods.

As Steel-Belted Radius attempts to authenticate a user, when it reaches a proxy entry on its **Authentication Methods** list, it generates a Proxy-radius request to the destination server in an effort to validate the username and password. If an accept is received from the proxy destination, Steel-Belted Radius will send an access-request for the user. If a reject is received from the destination proxy server, Steel-Belted Radius will check the next method on the authentication list.

Setting up proxy forwarding as an authentication method is useful for a number of reasons. For example:

- ◆ Proxy destinations can be set up without the need for the user to change anything about login. The user does not need to log into the proxy as **user@proxy**. The user can log in as **user** and Steel-Belted Radius can

automatically send the user's request to the proxy server to try and authenticate the user (to set this up, order the **<Proxy Name>** so that it appears above the **Native User** authentication method in the **Authentication Methods** list).

- ◆ Proxy authentication permits an easy migration path to Steel-Belted Radius from other RADIUS servers. The network administrator can install Steel-Belted Radius as the first RADIUS server, and set up proxy authentication to the old RADIUS server. Steel-Belted Radius will now receive all the access requests, and if the user cannot be authenticated against a Steel-Belted Radius database, then a proxy request will automatically be sent to the old RADIUS server.

External Databases

Steel-Belted Radius can use external SQL databases in conjunction with RADIUS authentication and accounting. Steel-Belted Radius can read an SQL database to retrieve authentication information, and it can write RADIUS accounting information to the database, independently of the Steel-Belted Radius accounting log.

When any external database is enabled as an authentication method, a corresponding entry appears in the Configuration dialog's **Authentication Methods** list, along with all the other methods available to this Steel-Belted Radius server. Each external database can be assigned a priority in this list, just like all the other methods.

As Steel-Belted Radius attempts to authenticate a user, when it reaches an external database name on its **Authentication Methods** list, it will try to select a record from the external database that matches the username and password provided in the user's access request. If it cannot find a matching record, Steel-Belted Radius will check the next method on the list.

External database authentication is normally used when an organization already has a large amount of user information stored in an SQL database, and wants 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 "by hand."

NOTE: An external database may be used for user authentication only. All other Steel-Belted Radius configuration information, such as RAS Clients, Profiles, IP

Pools, and so forth, must be entered and maintained in the Steel-Belted Radius database.

Configuring External Database Authentication

To set up an external database for use as a Steel-Belted Radius authentication method, you must place an **.aut** database configuration file in the Steel-Belted Radius server directory that you defined at installation time (usually **C:\RADIUS\Service**). This file must be modified to contain specialized information about your enterprise database.

*NOTE: See Chapter 7, “Using Steel-Belted Radius with SQL” for **.aut** file syntax, requirements, and examples.*

Configuring External Database Accounting

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 Steel-Belted Radius server directory that you defined at installation time (usually **C:\RADIUS\Service**). This file must be modified to contain specialized information about your enterprise database.

*NOTE: See Chapter 7, “Using Steel-Belted Radius with SQL” for **.acc** file syntax, requirements, and examples.*

Tunnels

Steel-Belted Radius supports the concept of a “tunnel,” a uniquely secure type of remote connection. This section provides background information about tunnels and explains how to configure the Steel-Belted Radius server to support them.

NOTE: If you don’t already use or plan to use tunnels on your network, then you don’t need the information in this section.

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 will request remote access and open the tunnel. An administrator at the remote site needs to 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 Network Access Server (NAS) at the remote site. The NAS retrieves the tunnel configuration attributes from its database and uses them to open the tunnel into the enterprise. Once the tunnel is open, the user can be authenticated at the enterprise.

Tunnels and the RADIUS Server

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. A RADIUS server that supports tunnels can:

- ◆ Determine whether the name string provided by the user includes a tunnel name (for example **user_name@tunnel_name**).
- ◆ Store lists of tunnels.
- ◆ 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.

The exact tunnel attributes stored on the RADIUS server may be different depending on the specific NAS hardware functionality. Steel-Belted Radius supports any set of database entries required to support the tunnel functionality of any of its supported RAS clients.

NOTE: Steel-Belted Radius does not give any RAS client capabilities that it doesn't already have (tunnels, IDSN support, and so on); Steel-Belted Radius simply offers compatible, centralized storage of the data required to use these capabilities.

Tunnel Connection Featuring RADIUS Servers

The following sequence highlights the role that RADIUS servers can play on both sides of a tunnel connection:

- 1 The user requests a connection by providing a name and password. The user may specify a name string that consists of:

user_name@realm_name

or:

realm_name@user_name

where **user_name** is the name of the dial-in user; @ is a delimiter character; and **realm_name** is the name of the target tunnel. The required order of names, and the required delimiter character, are configured by the RADIUS server administrator who sets up the tunnel. (For Steel-Belted Radius, tunnel configuration will be performed using the Administrator program Tunnels dialog.)

- 2 The remote site NAS passes the connection request to its RADIUS server for authentication.
- 3 The RADIUS server detects the tunnel delimiter character in the name string. It determines that this connection request involves a tunnel. Based on its tunnel name parsing rules, it extracts the **realm_name** portion of the name string input by the user.
- 4 If the RADIUS server recognizes the **realm_name** as a registered Tunnel name (for Steel-Belted Radius, this will be a Tunnel configured using the Tunnels dialog), then it retrieves the configuration data associated with this tunnel and returns it to the NAS with an Access Accept message.

*NOTE: Otherwise, the **realm_name** might indicate a proxy destination. For more information, see the topic “Proxy RADIUS” above.*
- 5 The NAS uses the tunnel configuration data to open a tunnel into the enterprise site. Authentication of the **user_name** will now be attempted, usually at the enterprise site. This authentication may employ a Steel-Belted Radius server, some other RADIUS server, or some entirely different authentication method.
- 6 If user authentication succeeds, the connection is complete. Otherwise, the user’s connection request is denied.

Configuring Tunnel Support

Steel-Belted Radius fully supports tunnels in concept. In practice, a specific Steel-Belted Radius server only supports the specific tunnels that have been

correctly configured on the system. This section explains how to perform the required configuration tasks to enable a tunnel connection.

To add a tunnel, you must open the Tunnels dialog in the Steel-Belted Radius Administrator program and add a Tunnel entry. In this entry, you will specify attributes such as the tunnel's name, its password, the IP address of the NAS on the "other side," encryption conventions to use, the order in which user and tunnel names are specified, the maximum number of tunnels that can be open at one time, and so on. You'll need to coordinate with the administrator on the other side to get some of this information.

Tunnel Name Parsing

To complete tunnel configuration, you must open the Configuration dialog in the Steel-Belted Radius Administrator program and choose a tunnel name parsing conventions. You can choose the suffix convention:

user_name@realm_name

or the prefix convention:

realm_name@user_name

You can choose a delimiter character other than @ (the default). Use care when setting the delimiter. You should use a different delimiter character for Tunnels than for Proxy entries, to avoid possible overlaps. If the delimiter character is the same for both, the Steel-Belted Radius server will be unable to tell whether **realm_name** is the name of a Tunnel or of a Proxy destination. To address this issue, when processing a connection request the server will always check its Tunnel entries first, then its Proxy entries.

NOTE: For more information, see the topic "Proxy RADIUS" above.

Tunnel Attributes

For each network vendor (Bay Networks, Shiva, etc.) there is a slightly different set of attributes or attribute values that can be set for a tunnel. The selections available to you in the Tunnels dialog will vary according to the dictionary of attributes that has been established for the specific type of NAS that will be opening this tunnel. The **Make/model** field of this NAS's RAS Client entry will determine which dictionary is used.

How Tunnels Use Dialed Number Information Services (DNIS)

DNIS (Dialed Number Information Services) refers to the capability of many Network Access Servers to determine and use the telephone number that was dialed to make a connection request. This information can be useful for authentication purposes.

DNIS Attributes

The RADIUS standard supports DNIS with attributes such as Calling-Station-Id (the number from which the user originated the request) and Called-Station-Id (the telephone number that was dialed to make the network connection).

DNIS and Tunnel Administration

Knowing which number was dialed to make the connection request can help determine whether this request requires a tunnel to be set up. Steel-Belted Radius uses the DNIS attribute Called-Station-Id to support tunnels as follows:

- ◆ When setting up a tunnel, the Steel-Belted Radius administrator can enter a telephone number or list of numbers in the Called-Station-Id list box on the Tunnels dialog.

NOTE: This list box does not set values for the RADIUS attribute of this name. Instead, it provides a list of expected real-time values for the attribute.
- ◆ After the Called-Station-Id list box has been set up for a tunnel, when the current connection request contains a Called-Station-ID value that indicates that a user has dialed into one of these listed telephone numbers, an association can be made to help determine whether or not a tunnel should be configured.

DNIS and Connection Requests

When a connection request is received, Steel-Belted Radius first attempts to determine from the username format whether or not the request is for a tunnel. If the username arrives as:

- ◆ **user_name@realm_name** or **realm_name@user_name**, Steel-Belted Radius will search its database for a tunnel entry that matches the **realm_name**. If a match can be found, then the information in the matching entry will be used to set up a tunnel.

NOTE: This tunnel will be set up even if the connection request does not contain a Called-Station-Id, or if its Called-Station-Id does not match any telephone number listed for the tunnel in the Steel-Belted Radius database.

If the **realm_name** does not match a tunnel entry in the database, then Steel-Belted Radius will check to see if the connection request contains a Called-Station-ID.

If so, Steel-Belted Radius will search its database for a tunnel entry that contains this phone number in its Called-Station-ID list. If a match can be found, then the information in the matching entry will be used to set up a tunnel.

Otherwise, Steel-Belted Radius continues checking other authentication types as usual.

- ♦ **username** only, Steel-Belted Radius will check to see if the request contains a Called-Station-ID. If so, Steel-Belted Radius will search its database for a tunnel entry that contains this phone number in its Called-Station-ID list. If a match can be found, then the matching entry will be used to set up a tunnel.

Otherwise, Steel-Belted Radius continues checking other authentication types as usual.

Resource Management

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

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 will be assigned 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 addresses to individual users. You can make fewer addresses go farther, and you can consolidate address assignment across all your RAS's.

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, it is essential that the RAS be configured for RADIUS accounting and that the same Steel-Belted Radius server be specified for both authentication and accounting.

If your RAS is not configured for accounting (or does not support accounting), then you cannot use the address pooling feature, because addresses would be assigned, but never released.

Setting Return-List Attributes for the User/Profile

The Framed-IP-Address (or Framed-IPX-Address) return-list attribute controls how the user's IP (or IPX) address will be assigned. For each user known to the Steel-Belted Radius Administration program, the Framed-IP-Address or Framed-IPX-Address attribute may be set in one of the following ways:

- ♦ **Static assignment.** Each time the user connects, the same specific address will be 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** will be assigned.
- ♦ **Assignment from a specific address pool.** When the user connects, an address will be assigned from a specific pool. For example, if user Kevin has Framed-IP-Address set to the **Sales** IP address pool, then when Kevin connects to the network, the next available IP address from **Sales** will be assigned.
- ♦ (For IP addresses only) **Assignment from the RAS Client's IP address pool.** When the user connects, an address will be assigned from the pool associated with the RAS client that makes the connection.

Let's say that RAS1 uses IP address pool **A**, RAS2 uses IP address pool **B**, and user Kevin has a Framed-IP-Address of **pool associated with RAS Client**. On connecting to the network, if user Kevin gets a port on RAS1, an IP address from pool **A** will be assigned. On the next call,

Kevin might connect to RAS2; in this case an address from pool **B** will be assigned.

Manually Releasing Addresses No Longer in Use

Normally, the system will take care of assigning and releasing addresses without any need to intervene. But under some circumstances, you can get “address leakage”; that is, an address is still being reserved for a user even 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 could occur for a variety of reasons:

- ♦ Steel-Belted Radius may have been taken down for a period of time during which accounting transactions occurred.
- ♦ The RAS may have been taken down or crashed before the user terminated. (In many cases, however, Steel-Belted Radius may be able to recover the addresses when the RAS starts up again.)
- ♦ The RAS may have sent the authentication and accounting transactions to a different RADIUS servers.
- ♦ Despite a successful authentication, the user’s PPP negotiation with the RAS may have terminated unsuccessfully for a variety of reasons. In such a case, some RAS’s may not initiate a subsequent accounting transaction.
- ♦ Routing problems may have prevented the accounting transaction from reaching Steel-Belted Radius.

An address that “leaked” will remain out of circulation until you manually release it, using the Statistics dialog, Current Users display.

Stopping and Starting the Server

Steel-Belted Radius maintains all current address assignments in a persistent database on disk. If you down the server and then restart it, all the information about which address is assigned to which user will be 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 have multiple IP or IPX address pools, it is perfectly permissible to duplicate some of the addresses among the pools. Steel-Belted Radius's address tracking mechanism ensures that if an IP address appears in more than one pool, once it is assigned out of any pool it will be unavailable through any of the pools until it is released.

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 Administration program allows you to limit the number of active connections per user, or per tunnel.

Concurrent User Connections

Steel-Belted Radius uses RADIUS Accounting information to determine the number of connections currently established for each user. You can use the Steel-Belted Radius Administration interface to set a maximum limit on the total number of concurrent connections that each user may have.

Then, when the user requests a new connection, Steel-Belted Radius can compare 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 Authentication log.

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

The maximum number of concurrent connections may be set individually for any user or group, of any type (**Native User**, **Domain User**, **Domain Group**, **Host User**, **Host Group**, **SecurID**, or **TACACS+**). For individual users, the limit will apply to the user; for groups, the limit will apply to all members of the group. For example, if **GroupA** has a connection limit of **2**, then users **\\GroupA\\userID1** and **\\GroupA\\userID2** will *each* be entitled to 2 concurrent connections.

The concurrent connection limit is set in the Users dialog by checking the **Maximum Concurrent Users** box and entering a number in the accompanying field.

For concurrent connection limits to work, it is essential that the RAS be configured for RADIUS accounting and that the same Steel-Belted Radius server be specified for both authentication and accounting.

If your RAS is not configured for RADIUS accounting (or does not support RADIUS accounting), then you cannot use the concurrent user connection feature. Without the accounting STOP record, Steel-Belted Radius will never know to release the user's connection.

Concurrent Tunnel Connections

Steel-Belted Radius allows you to limit the total number of simultaneous connections that can be open using a specific tunnel. This limit is controlled in the Steel-Belted Radius Administration interface with the Tunnels dialog checkbox **Maximum open tunnels**.

For concurrent tunnel connections to work, it is essential that the RAS that opens a tunnel be configured for RADIUS accounting and that the same Steel-Belted Radius server be specified for both authentication and accounting.

If your RAS is not configured for RADIUS accounting (or does not support RADIUS accounting), then you cannot use the concurrent tunnel connection feature. Without the accounting STOP record, Steel-Belted Radius will never know to close the tunnel connection.

Administering Steel-Belted Radius

4

The Steel-Belted Radius Administrator

The Steel-Belted Radius Administrator (**radadnt.exe**) is a Windows NT program that lets you easily and flexibly control all aspects of Steel-Belted Radius. In minutes you can set up new users, alter standard profiles, or configure new Remote Access Servers from any Windows NT machine on the network.

With the RADIUS Administrator you can:

- ♦ Select which Steel-Belted Radius Server to administer within your network.
- ♦ Configure one or more Network Access Servers (NAS's) as clients of Steel-Belted Radius.
- ♦ Authorize users or groups of users, set their connection attributes, and configure their access security.
- ♦ Set up standard connection profiles that can be included in any user profile.
- ♦ Grant authentication rights to another RADIUS server.
- ♦ Authorize tunnels and set up their connection attributes.
- ♦ Set up one or more pools out of which unique IP addresses will be assigned as users require them.
- ♦ Configure various operational characteristics of Steel-Belted Radius.
- ♦ View information on currently connected remote users across all NAS's.

Running the Steel-Belted Radius Administrator

To run the Steel-Belted Radius Administration program, double-click the RADIUS Administrator icon.

The Administration window appears and displays the Servers dialog.

The radio buttons at the left let you select which dialog to display. Just click the dialog you want and it will appear in the window. You must click **Servers** first and connect to a specific server before the other selections are enabled.

Getting Help with the Steel-Belted Radius Administrator

To get help with the Steel-Belted Radius Administrator, just press **[F1]** or select **Help Topics** from the Administrator menu.

To identify the current version of the Steel-Belted Radius Administrator, select **Help About** from the Administrator menu.

Exiting from the Steel-Belted Radius Administrator

To close the Steel-Belted Radius Administrator, select **File Exit** from the Administrator menu.

Closing the Steel-Belted Radius Administrator has no impact on the Steel-Belted Radius Server, which will continue to operate normally.

The Servers Dialog

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

To display the Servers dialog, check the **Servers** button at the left of the display.

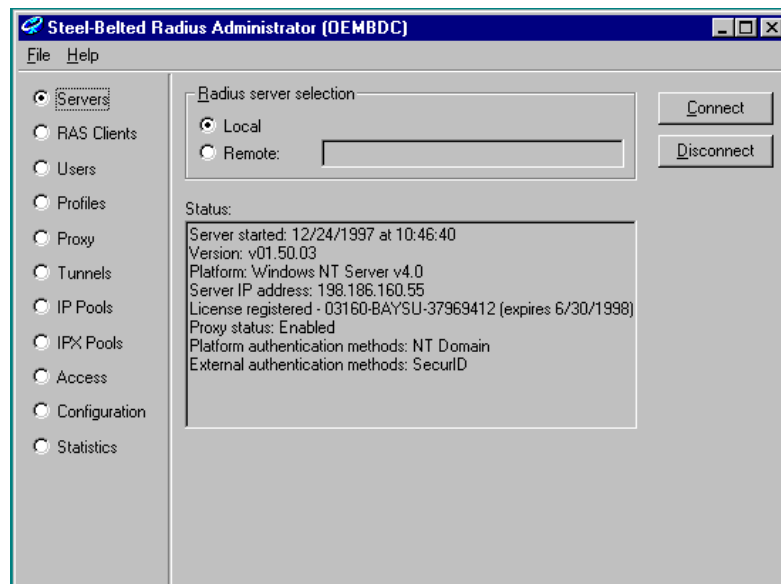


Figure 4-1: Servers Dialog

The **RADIUS Server Selection** box allows you to choose between the local RADIUS server or any other Windows NT machine running a RADIUS server.

To administer the RADIUS server running on the local Windows NT machine:

- ◆ Select **Local**, then click **Connect**.

To administer a RADIUS server running on another Windows NT machine:

- ◆ Select **Remote**, enter the name of the Windows NT machine, then click **Connect**.

Once connected, the **Status** box will list 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 may have occurred.

The RAS Clients Dialog

The RAS Clients dialog lets you identify the Remote Access Servers that may be clients of Steel-Belted Radius, and specify the IP address, shared secret, and other necessary information about each.

IMPORTANT: Steel-Belted Radius will not be able to communicate with a Remote Access Server that is not listed in this dialog.

To display the RAS Clients dialog, check the **RAS Clients** button at the left of the display.

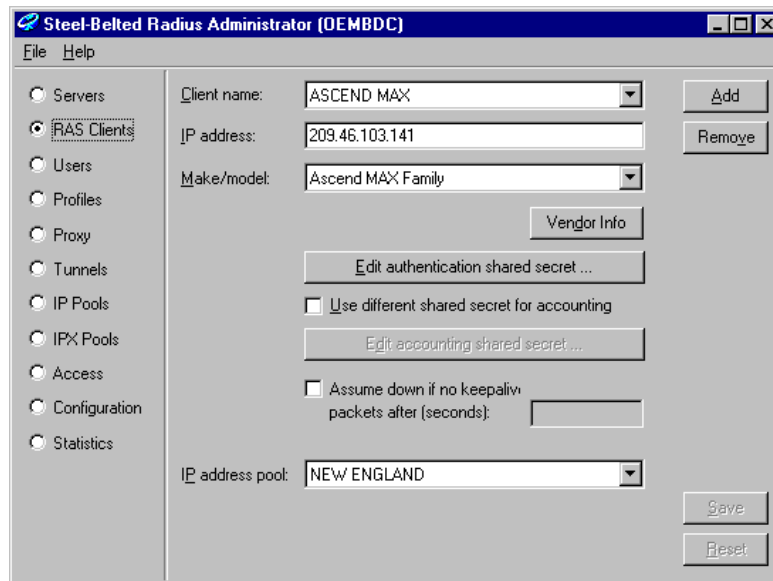


Figure 4-2: RAS Clients Dialog

Adding a New RAS Client

To add a new RAS Client:

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

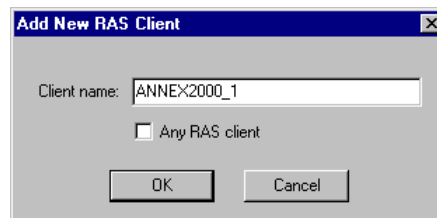


Figure 4-3: Add RAS Client Dialog

- 2 Enter the name of the RAS Client you'd like to add, and click **OK** to return to the RAS Clients dialog.

You will now see the name you entered in the Name field, with blank settings beneath.

NOTE: Although you can assign any name to the entry for a RAS, it is a good practice to use the RAS's actual name; this is normally the RAS's IP host name as well.

- 3 Edit the settings for the new RAS Client, as described below. Be sure you fill in all required fields.
- 4 Click **Save** to make your changes permanent.

Editing RAS Client Settings

The settings for each RAS Client include the RAS's IP address, its make and model, and a secret key that is shared between Steel-Belted Radius and the RAS.

Once you've edited any of 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

In order to communicate with the RAS, you must set its IP Address.

You can enter the IP address directly into the **IP Address** field.

Or, you can enter the DNS name of the RAS Client; the name you entered will be resolved and the IP address will be entered automatically into the **IP Address** field.

Make/model

The **Make/model** field selects the make and model of the RAS Client you are configuring.

Steel-Belted Radius has "personality" information for each RAS make and model in the list. This information includes a dictionary of check-list and return-list attributes, as well as other information that affects how the Steel-Belted Radius server communicates with this particular model of Remote Access Server.

To select the Make/model:

- 1 Click the **Make/model** drop-down box to bring up a list of the available RAS makes and models.

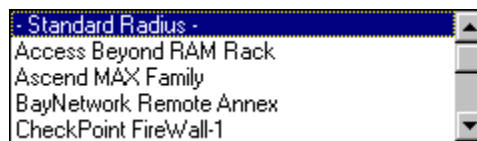


Figure 4-4: Make/model Selection List

- 2 Scroll through the list and select the item you wish; it will be displayed in the **Make/model** field. If you are not sure which Make/model you are using, or if your RAS is not in the list, you can select **Standard Radius**.

*NOTE: This information is extremely important. The Steel-Belted Radius server must know what type of RAS Client it is dealing with in order to properly communicate with that RAS. For more information about various brands of RAS Client supported by Steel-Belted Radius, click the **Vendor Info** button on the RAS Clients dialog.*

IP Address Pool

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

To associate the RAS 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.



Figure 4-5: Select IP Pool from List Dialog

- 2 Scroll through the list and select the item you wish; it will be displayed in the **IP Address Pool** field.

NOTE: A pool must be configured using the IP Address Pool dialog before Steel-Belted Radius will display it in this list.

Shared Secret

The Shared Secret is a critical component of communication between Steel-Belted Radius and the RAS. It is used to:

- ♦ Encrypt the password between the RAS and Steel-Belted Radius, when the RAS uses PAP (Password Authentication Protocol) authentication.
- ♦ Digitally sign each accounting request sent by the RAS to Steel-Belted Radius, so Steel-Belted Radius can verify its authenticity.
- ♦ Digitally sign each response sent by Steel-Belted Radius to the RAS, so the RAS can verify its authenticity.

While it is possible to use separate shared secrets for authentication and accounting requests between a RAS and a RADIUS server, it is more usual for the same shared secret to serve both purposes. The RAS Clients dialog permits you to accommodate either configuration.

To enter the authentication shared secret, click **Edit authentication shared secret** The Shared Secret dialog appears.

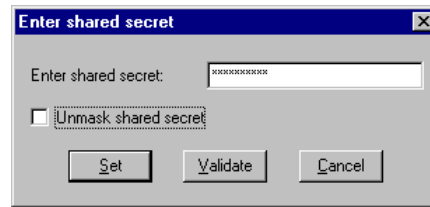


Figure 4-6: Shared Secret Dialog

To enter a shared secret, simply type it into the dialog and click **Set**.

For privacy, asterisks will be 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.

If you ever need to verify the shared secret, you can type it in and click **Validate**. You'll be told whether the shared secret is what you think it is.

If you need to use a separate shared secret for accounting, check **Use different shared secret for accounting**. Then click **Edit accounting shared secret ...** and enter the accounting shared secret into the pop-up dialog.

If **Use different shared secret for accounting** is not checked, the same shared secret will be used for both authentication and accounting.

***IMPORTANT:** The secret key(s) you enter here must correspond exactly to the secret key(s) that you entered into this RAS. If the keys don't match, Steel-Belted Radius and the Remote Access Server will not be able to communicate. Upper and lower case letters also make a difference!*

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 Steel-Belted Radius server has not received any RADIUS packets from this RAS after this number of seconds, it will assume that the RAS has gone down. Steel-Belted Radius will then gracefully close any user or tunnel connections through this RAS. That is, it will issue a STOP packet (which this RAS is presumably unable to do), release any

pooled IP or IPX addresses, and adjust the counts of concurrent user or tunnel connections appropriately.

*WARNING: If the **after (seconds)** value is set too small, valid user or tunnel connections may be lost. 60 seconds only lasts a minute; a user may be lost in thought and fail to generate packets for much longer than this.*

Adding a Wildcard RAS Client

A special RAS client entry, called **<ANY>**, allows Steel-Belted Radius to accept requests from any RAS Client 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 will 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 will accept 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.



Figure 4-7: RAS Client Selection List

- 2 In the RAS Clients dialog, click **Remove**.
- 3 You are prompted to confirm the operation.

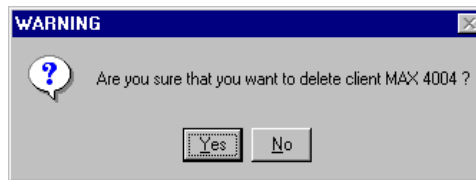


Figure 4-8: Remove RAS Client Confirmation Dialog

Click **Yes**.

The Users Dialog

The Users dialog lets you control which users may dial into the network, and specify their connection and security profiles.

To display the Users dialog, check the **Users** button at the left of the display.

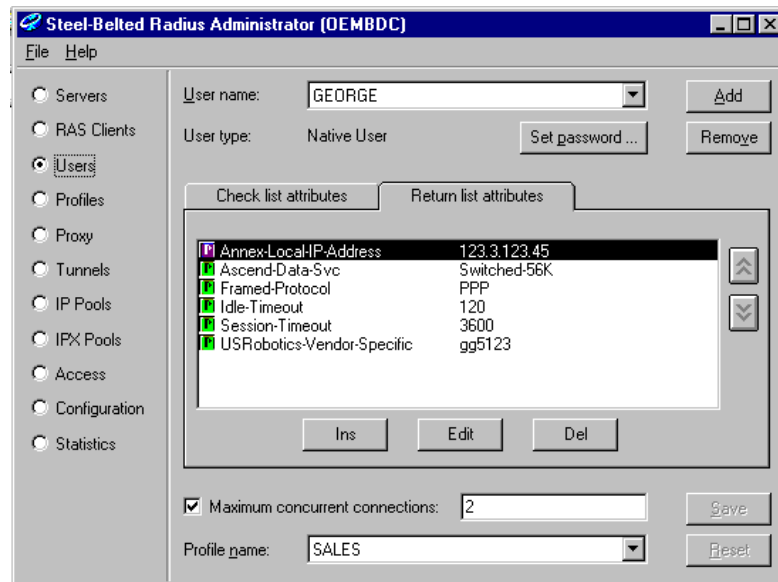


Figure 4-9: Users Dialog with Return-List Tab

The Users dialog **User type** field identifies which of the following authentication methods the Steel-Belted Radius server will apply to that user:

- ◆ A **Native User** is specified directly by entering the user's name and password for storage in the Steel-Belted Radius database.

- ♦ A **Domain User** is specified by selecting a Domain and a User name in that Domain.
- ♦ A **Domain Group** is specified by selecting a Domain and a Group within that Domain.
- ♦ A **Host User** is specified by selecting a Windows NT Host and a User name from that Host.
- ♦ A **Host Group** is specified by selecting a Windows NT Host and a Group from that Host.
- ♦ A **SecurID User** is specified by entering either the name of a specific user, or a prefix or suffix that allows a structured user name to be parsed, or a wildcard entry that allows any user to be authenticated via SecurID.
- ♦ A **TACACS+ User** is specified by entering either the name of a specific user, or a prefix or suffix that allows a structured user name to be parsed, or a wildcard entry that allows any user to be authenticated via TACACS+.

For each user dialing in to the network, Steel-Belted Radius will try each authentication method to see if the dial-in user can be authenticated.

NOTE: To set up the order in which the methods are tried, use the Configuration dialog.

If a Native User entry is being tried, the user is authenticated based on password information stored in the Steel-Belted Radius database. If a Domain or Host entry is being tried, the username and password are passed through to the Microsoft Windows Network for authentication. If a SecurID entry is being tried, the username, PIN, and token code are sent to an ACE/Server for authentication. If a TACACS+ entry is being tried, the username and password are sent to a TACACS+ Server for authentication. Depending on your configuration, the Steel-Belted Radius server may also check a proxy RADIUS server or an external database for information that will authenticate the user.

Once the appropriate user entry is found, Steel-Belted Radius uses the attribute information of that user entry to complete the request. If the check-list requirements are fulfilled, Steel-Belted Radius accepts the request and responds with the attributes in the return-list. If the check-list requirements are not fulfilled, or if no user entry is found against which the dial-in user can be authenticated, Steel-Belted Radius rejects the request.

Adding a Native User

A Native User is defined directly in the Steel-Belted Radius database. You may wish to define Native Users if you have non-Windows NT users who require remote access. For example, you can accommodate UNIX or Macintosh based users by adding them as Native Users.

To add a new Native User:

- 1 Click **Add**. The Add New User dialog appears.
- 2 Select the Native tab.

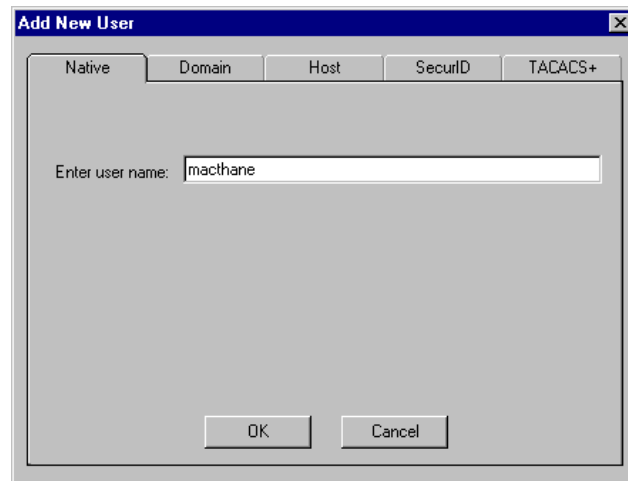


Figure 4-10: Add New Users Dialog with Native Tab

- 3 Enter the user's name and click **OK**. The Add New Users dialog closes.
Back in the Users dialog, you will now see the name you entered in the **User name** field, and the type of user in the **User type** field directly beneath the user name.
- 4 Click **Set Password**. Enter the password for the new user.
NOTE: The password is case-sensitive.
- 5 Edit the settings for the new user, as described below in "Editing User Settings."
- 6 Click **Save** to make your changes permanent.

Adding a Domain User or Domain Group

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 **Add**. The Add New User dialog appears.
- 2 Select the Domain tab.

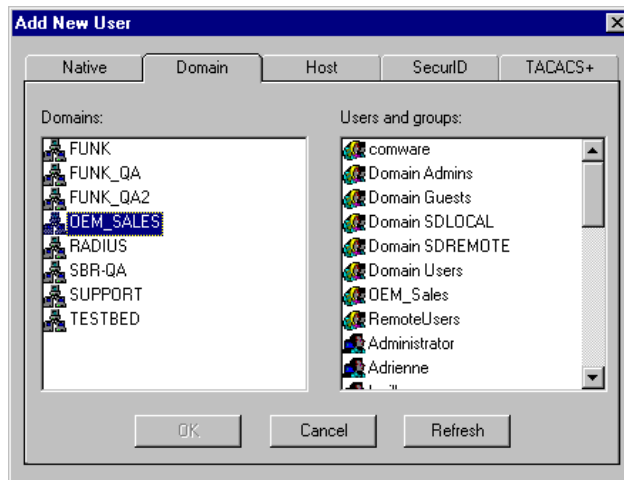


Figure 4-11: Add New Users Dialog with Domain Tab

- 3 First select a Domain name from the list at the left. Now select a user or group from the list of Domain objects at the right. Click **OK**.
You will now see the name you entered in the **User name** field, and the type of user in the **User type** field directly beneath the user name.
- 4 Edit the settings for the new user, as described below in “Editing User Settings.”
- 5 Click **Save** to make your changes permanent.

Adding a Host User or Host Group

Add a Host User entry to provide for the authentication of a specific user defined on a specific Windows NT 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 machine.

To add a new Host User or Host Group:

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

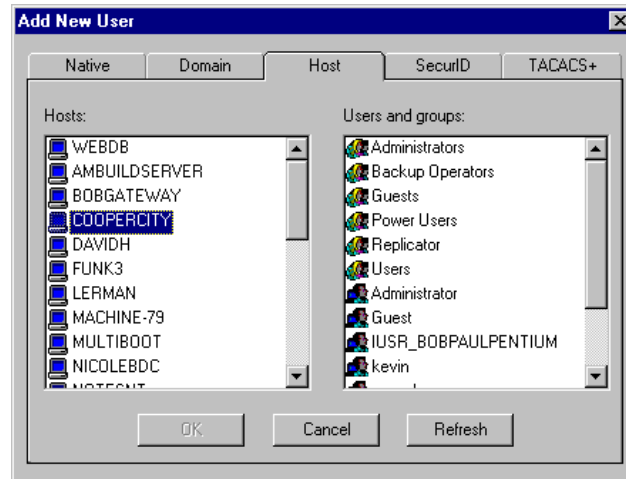


Figure 4-12: Add New Users Dialog with Host Tab

- 3 First select the name of an NT Host from the list at the left. Now select a Host user or group from the list on the right. Click **OK**.
You will now see the name you entered in the **User name** field, and the type of user in the **User type** field directly beneath the user name.
- 4 Edit the settings for the new user, as described below in “Editing User Settings.”
- 5 Click **Save** to make your changes permanent.

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.

SecurID authentication can coexist with other methods of authentication, with certain users authenticated via SecurID, and other users authenticated using other

methods. Steel-Belted Radius will try each authentication method in the order that it appears in the Configuration dialog.

Note that Steel-Belted Radius will only attempt SecurID authentication on usernames that match a SecurID user entry. There are four types of SecurID user entries, 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 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’d like to make SecurID either the only

authentication method used or the authentication method of last resort if prior authentication methods fail.

To add a new SecurID User entry:

- 1 Click **Add**. The Add New User dialog appears.
- 2 Select the SecurID tab.

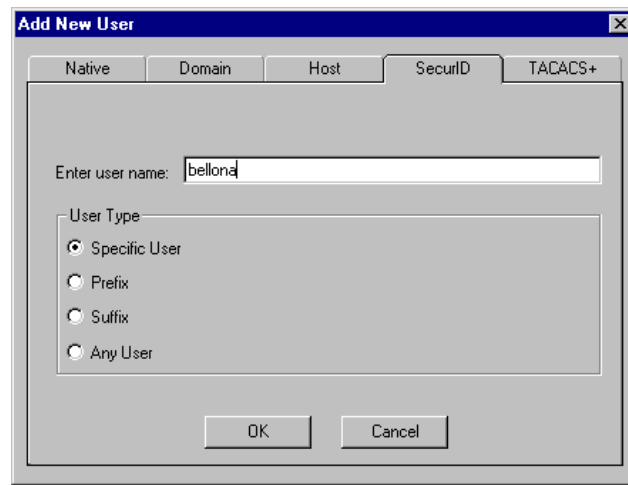


Figure 4-13: Add New Users Dialog with SecurID Tab

- 3 Select the user type; either **Specific user**, **Prefix**, **Suffix**, or **Any user**.
- 4 Enter the specific user name, prefix, or suffix. Click **OK**.

You will now see the name you entered in the **User name** field, and the type of user in the **User type** field directly beneath the user name.

- 5 Edit the settings for the new user, as described below in “Editing User Settings.”
- 6 Click **Save** to make your changes permanent.

Each new suffix or prefix entry that you add will appear 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.

TACACS+ authentication can coexist with other methods of authentication, with certain users authenticated via TACACS+, and other users authenticated using other methods. Steel-Belted Radius will try each authentication method in the order that it appears in the Configuration dialog.

Note that Steel-Belted Radius will only attempt TACACS+ authentication on usernames that match a TACACS+ user entry. There are four types of TACACS+ user entries, 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 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 prior authentication methods fail.

To add a new TACACS+ User entry:

- 1 Click **Add**. The Add New User dialog appears.
- 2 Select the TACACS+ tab.

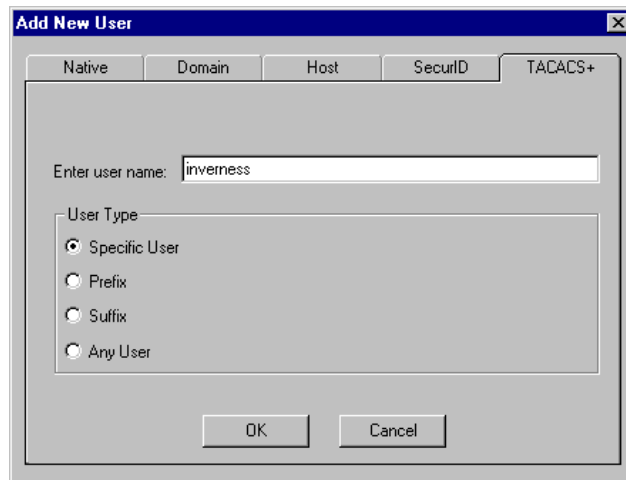


Figure 4-14: Add New Users Dialog with TACACS+ Tab

- 3 Select the user type; either **Specific user**, **Prefix**, **Suffix**, or **Any user**.
- 4 Enter the specific user name, prefix, or suffix. Click **OK**.
You will now see the name you entered in the **User name** field, and the type of user in the **User type** field directly beneath the user name.
- 5 Edit the settings for the new user, as described below in "Editing User Settings."
- 6 Click **Save** to make your changes permanent.

Each new suffix or prefix entry that you add will appear in the Users dialog with the username represented by the string **<USERNAME>**, for example **!<USERNAME>** or **sales\$<USERNAME>**.

Editing User Settings

The settings for each user entry include the user's password, check-list items, return-list items, and the name of the profile to use as a starting point for the check-list and return-list.

Once you've edited any of the user settings, click **Save** to make your changes permanent. If you change your mind and want to cancel your edits, click **Reset**.

Password

You can enter a password for Native users only. For all other user categories, Steel-Belted Radius does not need to keep passwords in its own database.

To enter a password for a Native user:

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

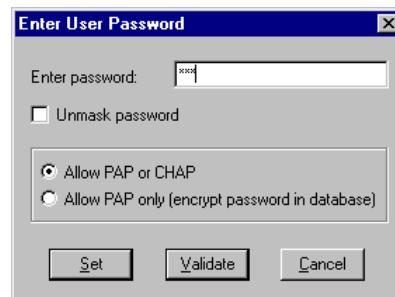


Figure 4-15: Enter User Password Dialog

- 2 Enter the password.
If you'd like the actual password to be echoed as you type rather than asterisks, check **Unmask password**.
- 3 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 will be totally secure even if your server is compromised.
- 4 Click **OK**.

NOTE: Steel-Belted Radius always encrypts the password it saves to its database. In order to use CHAP to authenticate the user, the Steel-Belted Radius

server uses a weaker form of encryption that enables it to recover the password in clear text as part of the CHAP implementation. PAP, however, allows the use of “strong encryption” (called a “one-way digest”) to store the password; this encryption makes it virtually impossible to recover the password from the information stored in the database.

Note that you can also use this dialog to verify a password you’ve already entered. Just type in a password and click **Validate**. You’ll be told whether the password is what you think it is.

Maximum Concurrent Users

The maximum number of concurrent connections may be set individually for any user or group, of any type (**Native User**, **Domain User**, **Domain Group**, **Host User**, **Host Group**, **SecurID**, or **TACACS+**). The limit is set in the Users dialog by checking the **Maximum Concurrent Users** box and entering a number in the accompanying field.

Profile

Steel-Belted Radius lets you define standard sets of check-list and return-list attributes called “profiles.” Any one of these profiles can be included in the settings for a user, rather than specifying attributes individually as described below.

Steel-Belted Radius profiles are similar to style sheets in a word processor. Rather than configuring different users with equivalent or similar attributes, you can define a set of attributes once in a profile, and simply apply that profile to each user. By using profiles, you can save typing, avoid errors, and you’ll be able to implement global changes without having to visit each user’s settings sheet.

When you include a profile in a user’s settings, you still have complete freedom to add attributes, or override profile attributes by modifying or removing them for the particular user.

You can create and modify profiles using the Profiles dialog, described below. We strongly recommend that you make use of this powerful feature, rather than separately entering each attribute for each user.

To select a profile to incorporate into the settings of a user:

- 1 Click the profile drop-down list.



Figure 4-16: Profile Selection List

- 2 Select the profile you'd like to use, or select **<no profile>**.

Check-List and Return-List Attributes

Check-list attributes indicate additional criteria beyond username and password for allowing a user access to the network. If you define check-list attributes for a user, then the RAS must send attributes to Steel-Belted Radius that match the check-list; if they don't, Steel-Belted Radius will issue a reject even if the user can be authenticated.

Return-list attributes are sent back to the RAS once authentication is successful and check-list requirements have been met. The return-list defines additional parameters that the RAS should assign to the connection, typically as part of PPP negotiations.

NOTE: Check-list and Return-list attributes are filtered based on the dictionary active for the make/model of RAS.

In most cases, determining how to set up attributes for the check-list and return-list shouldn't pose great difficulties. However, there may be occasions when you'll need to understand the full extent of the capabilities of the check-list and return-list.

Attribute Types

Each attribute consists of a name and a value. The value of each attribute has a well-defined type, which may be numeric, string, IP address, time, item in a list, or hexadecimal.

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

Multi-valued Attributes

Attributes may be single- or multi-valued; in other words, certain attributes may appear at most once in the check-list or return-list, while others may appear multiple 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 may 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 a Steel-Belted Radius 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 may 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 will retrieve the appropriate value for this attribute when it is needed.

The Echo Property

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

Suppose, for example, you add Callback-Number to the return-list and check **echo**. Steel-Belted Radius will take the value of the Callback-Number it receives in the RADIUS request and echo it back to the RAS in the RADIUS response; if it receives no Callback-Number, it echoes nothing.

Let us further suppose that 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 checking **default** for any check-list attribute, this indicates 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 possible use for default values would be to require that an attribute in a RADIUS request must take on 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 and is marked as default, and the same attribute appears in the return-list marked as echo, this means the following: If the attribute appears in the RADIUS request, echo it in the RADIUS response; however, if the attribute does not appear in the request, echo the default value in the response.

Note that 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 mark it as echo. Here's what happens: If a Callback-Number is present in the RADIUS request, it must match one of the check-list values or the user will be 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.

Attributes Inherited from a Profile

As noted earlier, check-list and return-list attributes can be directly entered for any user, or they may be inherited from the profile that has been selected as part of the user's settings.

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

Items that are inherited from the 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

Adding Check List Attributes

To add check-list attributes:

- 1 Click the Check List Attributes tab in the center of the dialog.
- 2 Click **Add**.
- 3 The Add New Attribute dialog appears.

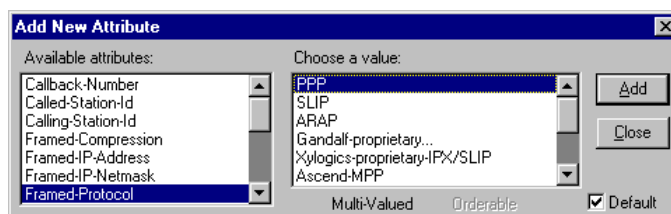


Figure 4-17: Add New Attribute Dialog

You will be able to 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.

- 4 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.
- 5 Enter a value for the attribute in the space to the right of the attribute list.
The method for entering a value varies with the type of attribute. You may need to enter a string, a numeric value, or select from a list of items.
- 6 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.
- 7 Click **Add** to add this attribute/value pair to the check list.
- 8 To add additional attributes, repeat steps 3 through 7 above until you are done.

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

Adding Return List Attributes

To add return-list attributes:

- 1 Click the Return List Attributes tab in the center of the dialog.
- 2 Click **Add**.
- 3 The Add New Attribute dialog appears.

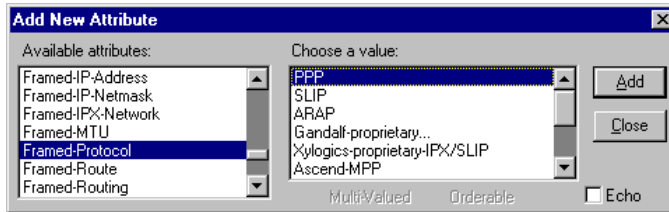


Figure 4-18: Add New Attribute Dialog

You will be able to 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.

- 4 Select an item from the list of **Available attributes**.
You'll be able to tell whether you can add multiple values for this attribute and whether you can control the order in which these values appear by noting the state of the **Multi-Valued** and **Orderable** indicators.
- 5 Enter a value for the attribute in the space to the right of the attribute list.
The method for entering a value varies with the type of attribute. You may need to enter a string, a numeric value, or select from a list of items.
- 6 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 RAS in the RADIUS response, check **Echo**.
*NOTE: **Echo** is only available for single-valued, Reply-List attributes.*
- 7 Click **Add** to add this attribute/value pair to the check list.
- 8 To add additional attributes, repeat steps 3 through 7 above until you are done.
- 9 Click **Close** to return to the User dialog.

Changing Attributes

To change the value of an attribute already in the check-list or return-list:

- 1 Click the Check List Attributes or Return List Attributes tab in the center dialog.
- 2 Highlight the attribute whose value you'd like to change.
- 3 Click **Edit** or double-click the attribute. A Change dialog displays with the attribute name in the title bar.

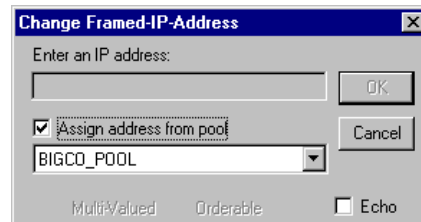


Figure 4-19: Change Attribute Dialog

Depending on the attribute, you may 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 Attributes

To remove an attribute/value pair from the check-list or return-list:



- 1 Click the Check List Attributes or Return List Attributes tab in the center dialog.
- 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 may appear more than once in a RADIUS response, and the order in which the attribute/value pairs appear is important.


To reorder attributes:


- 1 Highlight an attribute/value pair in the list.
- 2 Click one of the double-up and double-down arrows as follows:

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 grayed out.
	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 grayed out.

Changing Attributes Inherited from a Profile

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

To remove an attribute inherited from a profile, click **Remove**. The attribute will not disappear from the display, but will be shown grayed and struck-through, and will be 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 will be reset to its unmodified state, and will be marked with .

NOTE: If you accidentally delete a Profile in use as part of a User setting, you will 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.

Removing a User

To remove a user from the list:

- 1 Click the **Name** drop-down list and select the user you would like to remove.



Figure 4-20: User Selection List

-
- 2 In the Users dialog, click **Remove**. You will be prompted to confirm the deletion.

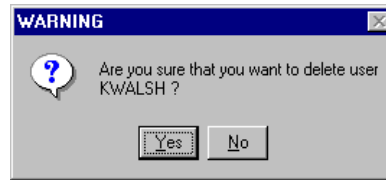


Figure 4-21: Remove User Confirmation Dialog

-
-
- 3 Click **Yes**. The user is removed from the list.

The Profiles Dialog

The Profiles dialog lets you define standard sets of check-list and return-list attributes. Any one of these profiles can be included in the settings for a user.

By defining commonly used sets of attributes in profiles, you can save yourself a lot of typing and avoid mistakes when you add new users. And if you'd like to change attributes settings across many users at once, you can do so easily just by changing the profile; the changes are automatically reflected in each user's settings.

Using a profile as part of a user's settings is in no way limiting. You can still make local changes to the profile for that user, such as adding new attributes, removing attributes, or changing attribute values. All the changes you make are simply local variations to the profile that applies just to this user and do not affect the profile definition itself.

To display the Profiles dialog, check the **Profiles** button at the left of the display.

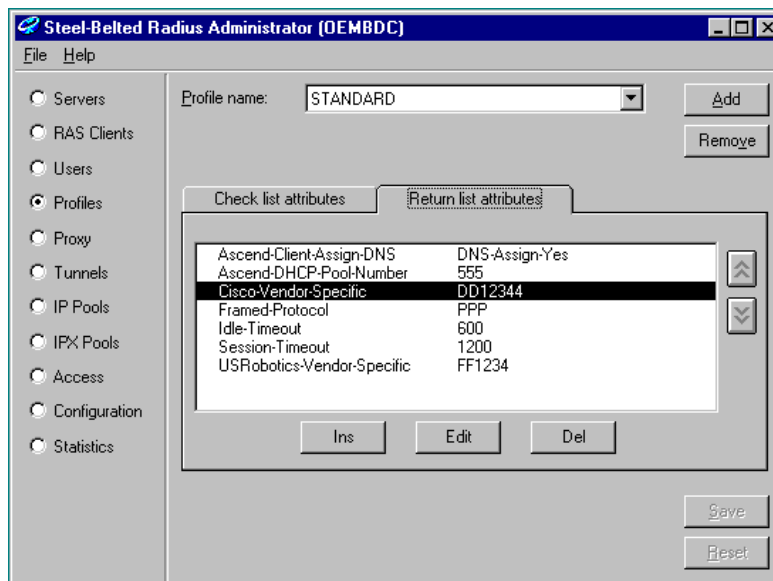


Figure 4-22: Profiles Dialog with Return-List Tab

Adding a Profile

To add a new profile:

- 1 Click **Add**. The Add New Profile dialog appears.

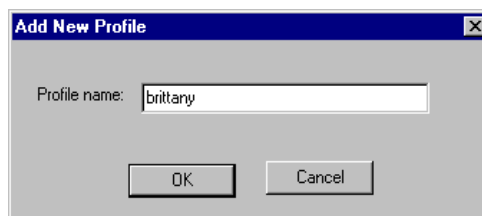


Figure 4-23: Add New Profile Dialog

- 2 Enter a name for the new profile, and click **OK** to return to the Profiles dialog.

You will now see the name you entered in the Profiles dialog **Name** field, with an empty attribute list below.

- 3 Add check-list and return-list attributes for the new profile.

For complete information on attributes and how to add them, please refer to the description of “Check-List and Return-List Attributes” in the “Users Dialog” section.

- 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.

For complete information on attributes and how to edit them, please refer to the description of “Check-List and Return-List Attributes” in the “Users Dialog” section.

Removing a Profile

To remove a Profile from the list:

- 1 Click the **Name** drop-down list and select the Profile you would like to remove.



Figure 4-24: Profile Selection List

- 2 In the Profiles dialog, click **Remove**. You will be prompted to confirm the deletion.

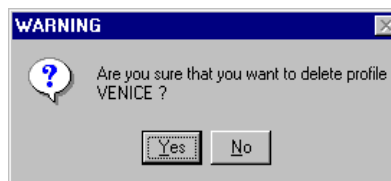


Figure 4-25: Remove Profile Confirmation Dialog

- 3 Click **Yes**. The profile is removed from the list.

WARNING: Be sure you don't remove a Profile that is currently included in the settings of a user. You will be warned that the profile name is in use by one or more user entries. If you delete the profile anyway, the attributes defined in the

Profile will disappear from that user's settings; when you next display the user's settings, you will get an error message asking you to edit and resave those settings.

The Proxy Dialog

The Proxy dialog lets you configure Steel-Belted Radius to act as a proxy server, and forward requests to other RADIUS servers.

Steel-Belted Radius determines the target RADIUS server by parsing the username it receives from the RAS. If the username is of the form **user_name@target_name**, Steel-Belted Radius will forward the request to the specified target server (**target_name**).

For each target RADIUS server to which requests may be forwarded, you must add an entry in the Proxy dialog.

You can also add a special **<ROAMING>** entry, to allow forwarding to any target with matching shared secret. If you include the **<ROAMING>** entry, if a target name is received that does not have a corresponding entry in the database of proxy targets, a DNS lookup is performed on that target name to determine where to forward it.

Any target RADIUS server that you configure using the Proxy dialog may be used for proxy authentication. You can enable a proxy as an authentication method simply by checking a box in the Proxy dialog. You must then open the Configuration dialog to establish a priority for this proxy server in the **Authentication Methods** list, relative to the other authentication methods such as **Native User** or **SecurID**.

NOTE: See Chapter 3, "RADIUS Concepts," for details about proxy servers.

To configure a proxy server, check the **Proxy** button at the left of the display.

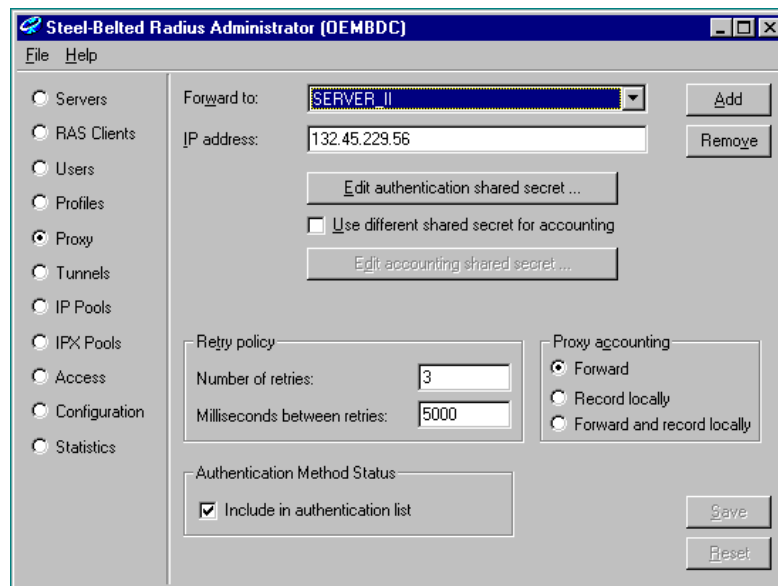


Figure 4-26: Proxy Dialog

Adding a New Target Server

To add a new target proxy server:

- 1 Click **Add**. The Add New Target Server dialog appears.



Figure 4-27: Add New Target Server Dialog

- 2 Enter the name of the target server you'd like to add, and click **OK**.

You will now see the name you entered in the **Forward to** field, with blank settings beneath.

***IMPORTANT:** Be sure the name you enter matches the target name that appears after the @ in the username; if Steel-Belted Radius receives a username with a name it does not recognize, it will reject the request.*

- 3 Edit the settings for the new target server entry, as described below. Be sure you fill in all required fields.
- 4 Click **Save** to make your changes permanent.

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.

Once you've edited the settings, click **Save** to make your changes permanent. If you change your mind and want to cancel your edits, click **Reset**.

IP Address

In order to communicate with the target server, you must set its IP Address.

You can enter the IP address directly into the **IP Address** field.

Or, you can enter the DNS name of the target server; the name you entered will be resolved and the IP address will be entered automatically into the **IP Address** field.

Shared Secret

The shared secret between a proxy and target RADIUS server serves the same purpose as the one between a RAS and a RADIUS server. The shared secret is used to encrypt password information and to authenticate one server to the other.

As discussed previously, it is possible to use separate shared secrets for authentication and accounting requests, but it is more usual for the same shared secret to serve both purposes. The Proxy dialog permits you to accommodate either configuration.

To enter the authentication shared secret, click **Edit authentication shared secret ...**. The Shared Secret dialog appears.

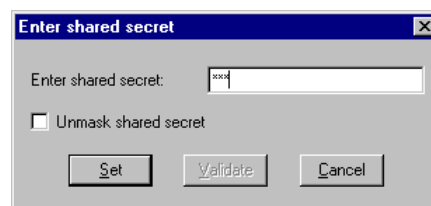


Figure 4-28: Shared Secret Dialog

To enter a shared secret, simply type it in to the dialog and click **Set**.

For privacy, asterisks will be 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.

If you ever need to verify the shared secret, you can type it in and click **Validate**. You'll be told whether the shared secret is what you think it is.

If you need to use a separate shared secret for accounting, check **Use different shared secret for accounting**. Then click **Edit accounting shared secret ...** and enter the accounting shared secret into the pop-up dialog.

Note that if **Use different shared secret for accounting** is not checked, the same shared secret will be used for both authentication and accounting

***IMPORTANT:** The secret key(s) you enter here must correspond exactly to the secret key(s) that you enter into the target RADIUS server's RAS Clients entry for this proxy server. If the keys don't match, this proxy server and the target server will not be able to communicate. Upper and lower case letters make a difference!*

Retry Policy

When Steel-Belted Radius acts as a proxy, it needs to emulate the characteristics of a RAS. This includes the ability to retransmit a request if it doesn't get a response within some interval of time. There are two values which can be set:

- ♦ **Number of Retries.** This sets the number of times a request will be retransmitted in case an acknowledgment from the target is not received; if the number of retries is exhausted, then the original request will be 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 control whether or not the proxy server that you are defining can be selected and ordered as one of several **Authentication Methods** on the Configuration dialog. To enable this feature, check the **Include in authentication list** box.

NOTE: See the Chapter 3 topic “Proxy RADIUS” for background information.

Adding Roaming Proxy Support

To add a **<ROAMING>** entry to the database of target servers:

- 1 Click **Add**. The Add New Target Server dialog appears.
- 2 Check **Roaming** and click **OK**.

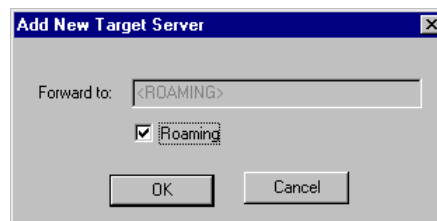


Figure 4-29: Adding a **<ROAMING>** Proxy Target

You will now see **<ROAMING>** in the **Forward to** field, with blank settings beneath.

- 3 Update the **Shared Secret**, **Retry Policy**, and **Proxy Accounting** fields for this item, then click **Save** to make your changes permanent.

Note that the IP Address field cannot be edited. **<ROAMING>** implies that the server will determine the IP address of the target server by resolving the target name using DNS.

Removing a Target Server

To remove a target server from the list:

- 1 Click the **Forward** to drop-down list and select the target server you'd like to remove.
- 2 In the Proxy dialog, click **Remove**.

The Tunnels Dialog

NOTE: If you don't already use or plan to use tunnels on your network, then you don't need the information in this section.

The Tunnels dialog lets you configure Steel-Belted Radius to support a tunnel, a uniquely secure type of remote connection. Other configuration tasks and background information are necessary to successfully complete tunnel setup.

NOTE: See Chapter 3, "RADIUS Concepts," for background information.

To add or remove a tunnel entry in the Steel-Belted Radius database, check the **Tunnels** button at the left of the display.

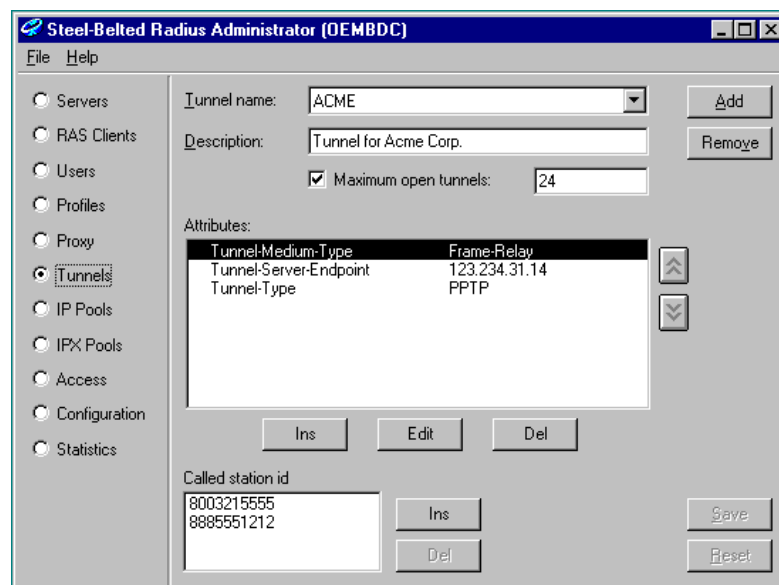


Figure 4-30: Tunnels Dialog

Adding and Editing a Tunnel

To add a tunnel entry:

- 1 At the Tunnels dialog, click **Add**.

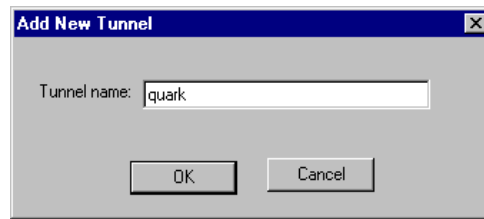


Figure 4-31: Add New Tunnel Dialog

- 2 Enter the tunnel name and click **OK**.
- 3 Set the **Maximum open tunnels** number. This is the maximum number of simultaneous open connections that your site can have using this tunnel.
- 4 Enter a text **Description** of the tunnel, for example the name of the organization that will use it. This text is for administrative use only and does not affect tunnel connections.
- 5 Each Tunnel definition includes various attributes. The specific attributes available in the Tunnels dialog depend on the **Make/model** of the RAS Client you are using. You can add, modify, and remove attributes in the Tunnels dialog using similar techniques as in other dialogs.

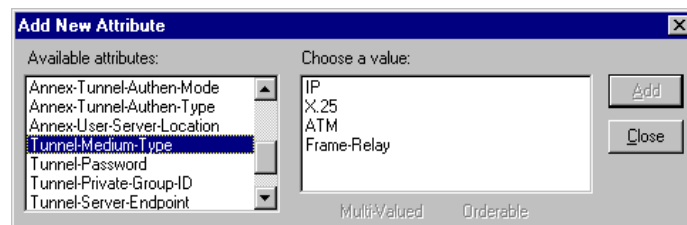


Figure 4-32: Add New Attribute Dialog (for a Tunnel)

For complete information on attributes and how to edit them, please refer to the description of “Check-List and Return-List Attributes” in the “Users Dialog” section.

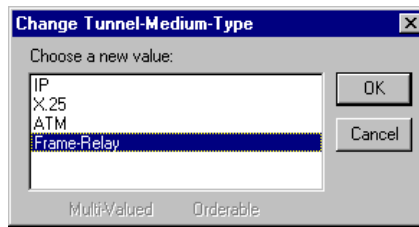


Figure 4-33: Change Attribute Value Dialog (for a Tunnel)

- 6 Click **Save** to make your changes permanent, **Reset** to undo them.

Removing a Tunnel

To remove a Tunnel from the list:

- 1 Click the **Name** drop-down list and select the Tunnel you would like to remove.



Figure 4-34: Tunnel Selection Dialog

- 2 In the Tunnels dialog, click **Remove**. You will be prompted to confirm the deletion.

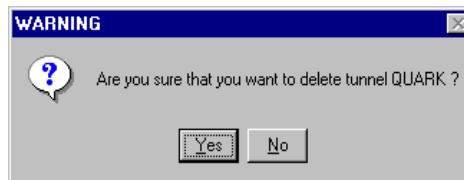


Figure 4-35: Remove Tunnel Confirmation Dialog

- 3 Click **Yes**. The tunnel is removed from the list.

The IP Pools Dialog

The IP Pools Dialog allows you to set up one or more pools out of which unique IP addresses will be assigned as users require them. Each pool consists of a list of one or more ranges of IP addresses.

IMPORTANT: Before committing yourself to using pooled IP address assignment, be sure to read “How Address Assignment Works” in the RADIUS Concepts chapter. Depending on your overall configuration, certain limitations may apply to your use of this feature.

To display the IP Pools dialog, check the **IP Pools** button at the left of the display.

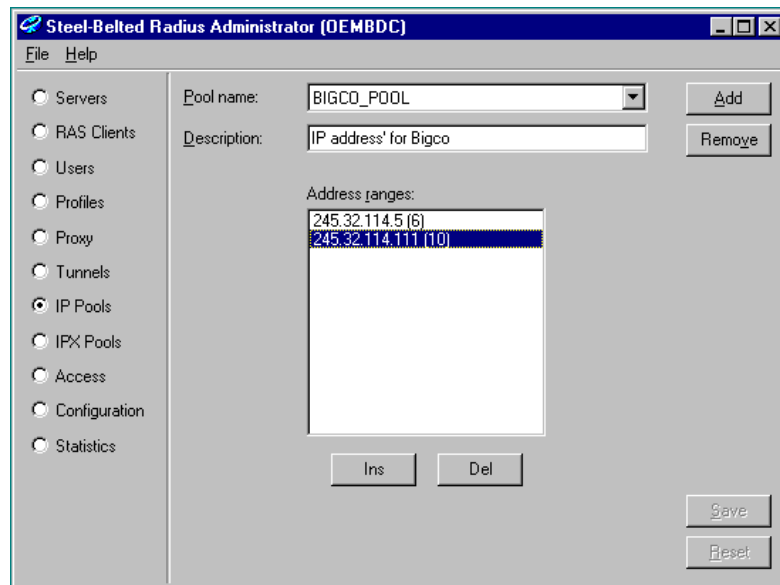


Figure 4-36: IP Pools Dialog

Adding an IP Address Pool

An IP address pool consists of one or more ranges of IP addresses. You can also set an optional description for each address pool.

To add a new IP address pool:

- 1 Click **Add**. The Add New IP Address Pool dialog appears.

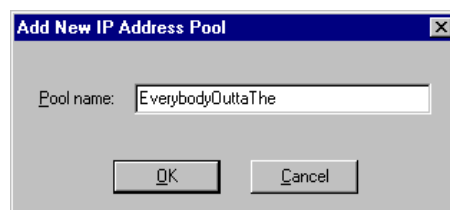


Figure 4-37: Add New IP Address Pool Dialog

- 2 Enter a name for the new address pool, and click **OK** to return to the IP Pools dialog.
You will now see the name you entered in the **Pool name** field, with a list of address ranges below.
- 3 Set the description for this address pool, and add ranges of IP addresses that make up the pool, as described below.
- 4 Click **Save** to make your changes permanent.

Editing an IP Address Pool

To add a new range of IP addresses:

- 1 Click **Ins**. The Add New IP Address Range dialog appears.

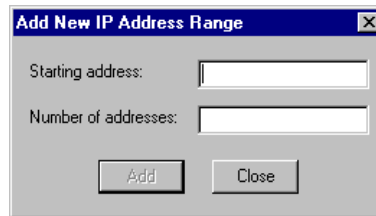


Figure 4-38: Add New IP Address Range Dialog

- 2 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.
- 3 When done adding ranges, click **Close** to return to the IP Pools dialog.

To remove a range of IP addresses from the pool, highlight the address range you'd like to remove and click **Del**.

Removing an IP Address Pool

To remove an IP Pool from the list:

- 1 Click the **Pool name** drop-down list and select the IP Pool you would like to remove.



Figure 4-39: Select IP Address Pool Dialog

- 2 In the IP Pools dialog, click **Remove**.
- 3 You will be prompted to confirm the deletion.

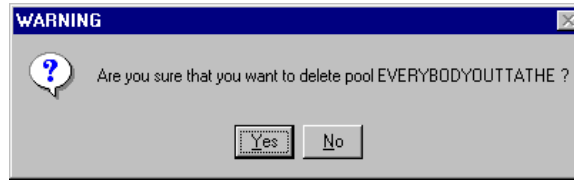


Figure 4-40: Remove IP Address Pool Confirmation Dialog

Click **Yes**. The IP pool is deleted.

Specifying IP Address Assignment in User/Profile Records

The Framed-IP-Address return-list attribute controls how the Steel-Belted Radius server will assign 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.

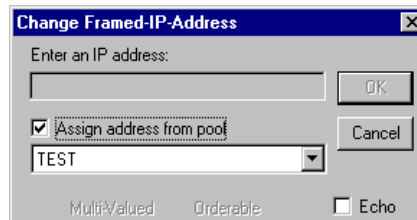


Figure 4-41: Editing the Framed-IP-Address Attribute

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. The last selection on the list is **pool associated with RAS client**; if you select this option, make sure you've used the RAS Clients dialog to associate each RAS with a specific IP Address Pool.

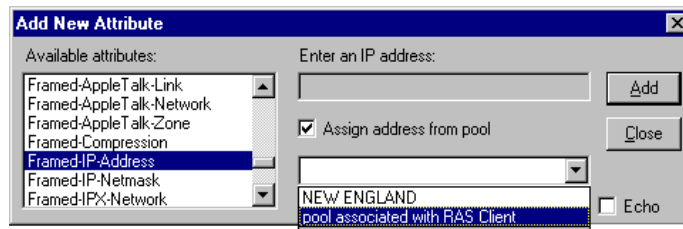


Figure 4-42: Specifying an IP Pool for the Framed-IP-Address Attribute

The IPX Pools Dialog

The IPX Pools Dialog allows you to set up one or more pools out of which unique IPX network numbers will be assigned as users require them. Each pool consists of a list of one or more ranges of IPX network numbers.

IMPORTANT: Before committing yourself to using pooled IPX network number assignment, be sure to read “How Address Assignment Works” in the RADIUS Concepts chapter. Depending on your overall configuration, certain limitations may apply to your use of this feature.

To display the IPX Pools dialog, check the **IPX Pools** button at the left of the display.

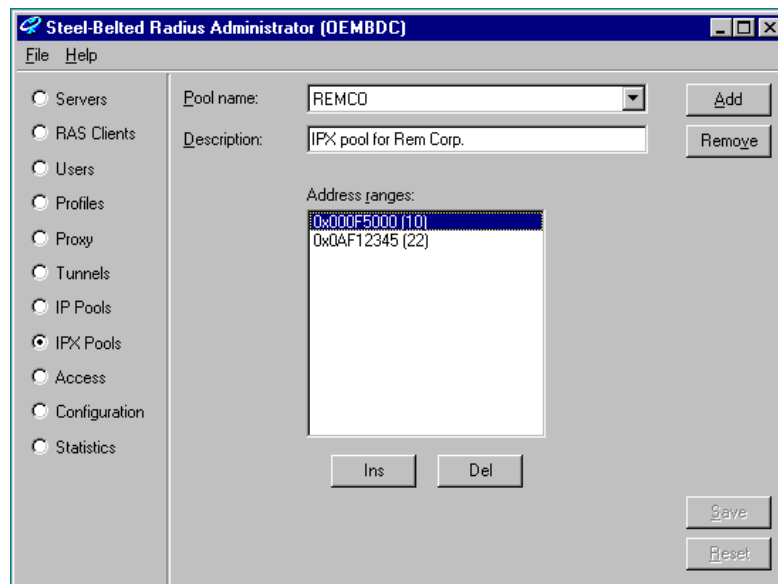


Figure 4-43: IPX Pools Dialog

Adding an IPX Pool

An IPX pool consists of one or more ranges of IPX network numbers. You can also set an optional description for each address pool.

To add a new pool of IPX network numbers:

- 1 Click **Add**. The Add New IPX Address Pool dialog appears.

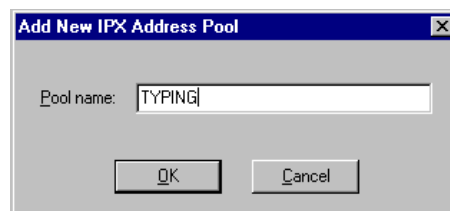


Figure 4-44: Add New IPX Address Pool Dialog

- 2 Enter a name for the new address pool, and click **OK** to return to the IPX Pools dialog.

You will now see the name you entered in the **Pool name** field, with a list of address ranges below.

- 3 Set the description for this address pool, and add ranges of IPX network numbers that make up the pool, as described below.
- 4 Click **Save** to make your changes permanent.

Editing an IPX Pool

To add a new range of IPX network numbers:

- 1 Click **Ins**. The Add New IPX Address Range dialog appears.

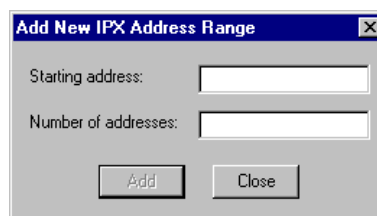


Figure 4-45: Add New IPX Address Range Dialog

- 2 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.

- 3 When done adding ranges, click **Close** to return to the IPX Pools dialog.

To remove a range of IPX network numbers from the pool, highlight the address range you'd like to remove and click **Del**.

Removing an IPX Pool

To remove an IPX Pool from the list:

- 1 Click the **Pool name** drop-down list and select the IPX Pool you would like to remove.



Figure 4-46: IPX Pool Selection List

- 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 will assign 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.

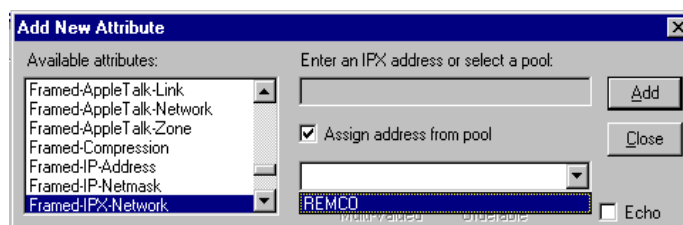


Figure 4-47: Specifying an IPX Pool for the Framed-IPX-Address Attribute

The Access Dialog

Anyone using the Steel-Belted Radius Administrator program must have specific rights. The Access dialog lets you grant and revoke these rights.

Anyone that has “Administrative” rights (that is, anyone who is a member of the Windows NT Administrators group) has implicit rights to use the Steel-Belted Radius Administrator. If you are content to let only users with Administrative rights use the Steel-Belted Radius Administrator program, you do not need to use the Access dialog at all to grant rights.

To display the Access dialog, check the **Access** button at the left of the display.

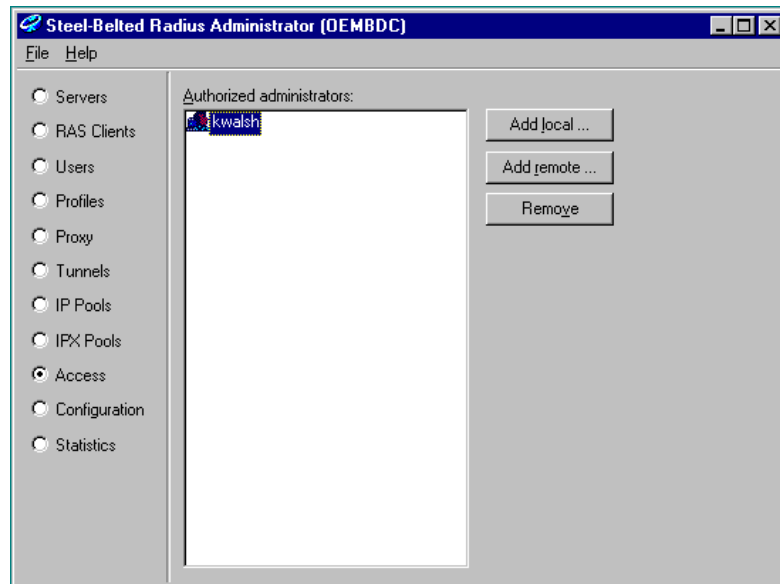


Figure 4-48: Access Dialog

The **RADIUS Administrators** list show the users and groups that have been explicitly granted the right to administer the Steel-Belted Radius server.

Local users or groups will be shown with their normal name. **Remote** users or groups will be shown with the name of the Domain, followed by a backslash and then the name of the Domain user or group.

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.

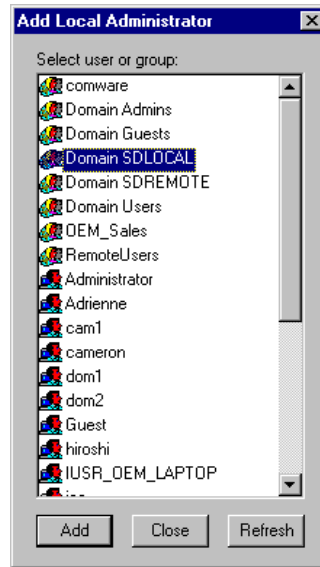


Figure 4-49: Add Local Administrator Dialog

- 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.

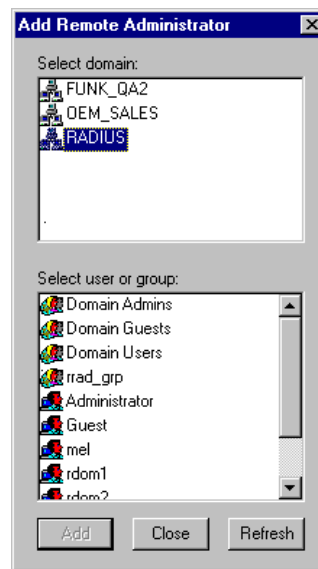


Figure 4-50: Add Remote Administrator Dialog

- 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**.

***WARNING:** Be careful not to revoke your own rights. If you do so, you will no longer have access to RADIUS administrative functions.*

The Configuration Dialog

Steel-Belted Radius lets you control many aspects of the behavior of the server. Specifically, you can configure the following elements:

- ♦ The text of various reject messages that will be sent to the RAS client (and possibly to the user) when the RADIUS request is rejected.
- ♦ The order in which different methods of authentication will be attempted.
- ♦ The number of days that the RADIUS server should retain authentication and accounting logs before the files are recycled.
- ♦ How the server should parse tunnel names: Should the realm name or user name be first, and which character should separate the names?

To display the Configuration dialog, check the **Configuration** button at the left of the display.

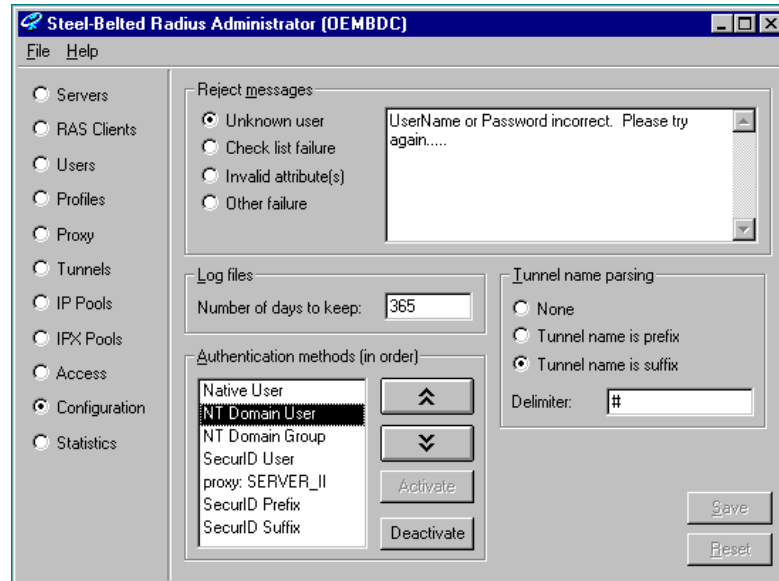


Figure 4-51: Configuration Dialog

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 will take effect.

To revert to the previous settings, click **Reset**.

Reject Messages

Steel-Belted Radius can provide different types of error messages to a RAS requesting authentication. Based on the type of error, a different Reject Message will be returned. You can configure the message text that will be returned to the RAS when a particular type of error occurs.

Reject Message Types

The following table describes the Reject Messages and their meaning to the RAS or dial-in user.

Message	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 Attributes	The request contained an attribute in violation of the RADIUS specification.
Other	Some other error occurred, such as a resource failure.

Changing Reject Messages

To modify Reject Message text:

- 1 Select a Reject Message type. The message text displays to the right.
- 2 In the message text display field, edit the text or type the new message.

Tunnel Name Parsing

A user who attempts a connection provides a name string and a password string. A name string can have multiple parts: the name of the dial-in user; a delimiter character, usually @; and the name of a target tunnel. You configure the required order of names, and the required delimiter character, in the Configuration screen. Steel-Belted Radius will use this information to parse incoming user connection requests and interpret them as tunnels, proxies, or Native users.

The name string can use a **tunnel suffix**:

user_name@tunnel_name

or a **tunnel prefix**:

tunnel_name@user_name

If you select **None**, tunnel name parsing is disabled and the Steel-Belted Radius server will not be able to support tunnels.

Authentication Methods Configuration

Steel-Belted Radius can authenticate a username/password using any of the following methods: **Native User**, **Domain User**, **Domain Group**, **Host User**, **Host Group**, **SecurID**, **TACACS+**, **Proxy RADIUS**, or **External Database**.



When Steel-Belted Radius receives a user name, it does not know in advance to which authentication category this user will belong. It must try each method.

The **Authentication Methods** selection allows you to fine-tune the sequence of authentication attempts. You can:

- ◆ Specify the order in which different methods of authentication will be performed.
- ◆ Disable one or more methods of authentication.

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 grayed out.
	Moves the selected method down one slot in the list. If the selected method is already last in the list, then the button is grayed out.

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 on the file server.

There are separate logs for authentication transactions and accounting transactions. Each day at midnight, the previous day's log files are completed, and new authentication and accounting log files are created for the new day's transactions.

In order to prevent the log files from continuously depleting available disk space Steel-Belted Radius will retain the log files for some period of time and then automatically delete them.

To specify the number of days to retain log files, enter a value in the **Days to Keep** field.

The Statistics Dialog

Steel-Belted Radius provides information on the status of the server. The Statistics dialog lets you:

- ♦ View statistics related to the authentication service
- ♦ View statistics related to the accounting service
- ♦ View statistics related to proxy forwarding
- ♦ View the amount of time Steel-Belted Radius has been running
- ♦ View an on-screen report of all users currently connected to a Remote Access Server, based on real-time RADIUS accounting information

To display the Statistics dialog, click the **Statistics** button at the left of the display. The Statistics Dialog provides three tabs with statistics for all types of RADIUS activity on the currently selected server: Authentication, Accounting, and Proxy.

Authentication Statistics

Authentication statistics provide information such as the number of accept and reject messages and the reasons for rejecting authentication.

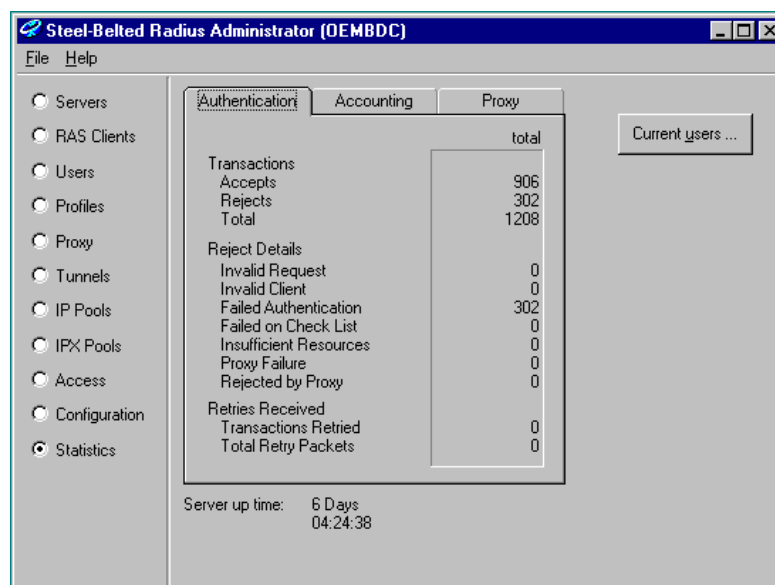


Figure 4-52: Statistics Dialog with Authentication Tab

The following table describes the authentication statistics, with possible interpretations in *italics*.

Authentication Statistic	Meaning
Transactions	
Accepts	The total number of RADIUS transactions resulting in an accept response.
Rejects	The total number of RADIUS transactions resulting in a reject response. These are broken out in Reject Details below.
Silent Discards	<p>The total number of requests in which the RAS Client could not be identified.</p> <p><i>A RAS may have been configured to use Steel-Belted Radius but no RAS Client entry has been created with the name and/or IP address of the RAS; or the RAS Client entry has been misconfigured with an incorrect name or IP address; or some rogue device is attempting to compromise RADIUS security.</i></p>
Reject Details	
Invalid Request	<p>The total number of invalid RADIUS requests made.</p> <p><i>A RAS is sending incorrectly formed packets to Steel-Belted Radius; either there is a configuration error or the RAS does not conform to the RADIUS standard.</i></p>
Failed Authentication	<p>The total number of failed authentication requests, where the failure is due to invalid username or password.</p> <p><i>If all transactions are failing authentication, then the shared secret entered into Steel-Belted Radius may not match the secret entered on the RAS since it will be decrypted incorrectly.</i></p>
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.
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 will rarely match, as many transactions may be “in progress” at any given time.

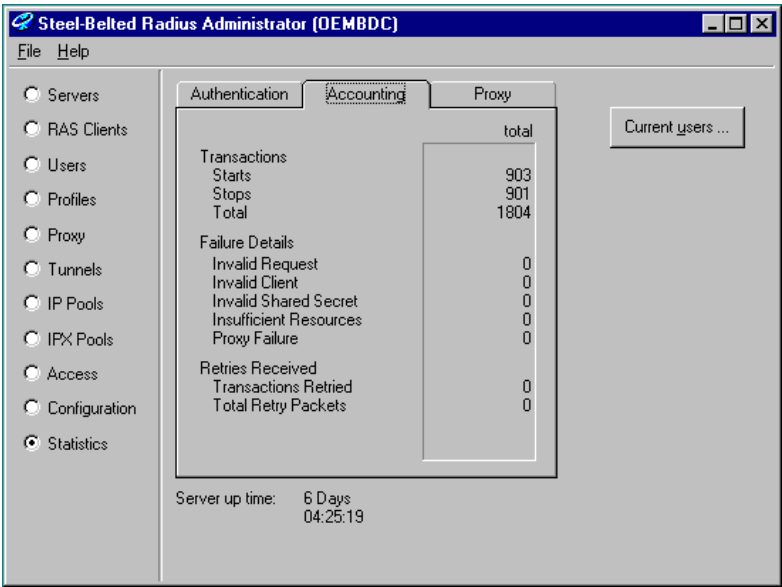


Figure 4-53: Statistics Dialog with Accounting Tab

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.
Failure Details	
Invalid Request	<p>The total number of invalid RADIUS requests made.</p> <p><i>A RAS is sending incorrectly formed packets to Steel-Belted Radius; either there is a configuration error or the RAS does not conform to the RADIUS standard.</i></p>
Invalid Client	<p>The total number of requests in which the RAS Client could not be identified.</p> <p><i>A RAS may have been configured to use Steel-Belted Radius but no RAS Client entry has been created with the name and/or IP address of the RAS; or the RAS Client entry has been misconfigured with an incorrect name or IP address; or some rogue device is attempting to compromise RADIUS security.</i></p>
Invalid Shared Secret	<p>The total number of packets for which an incorrect digital signature was received.</p> <p><i>The shared secret does not match between Steel-Belted Radius and the RAS; or some unauthorized rogue device is attempting to compromise RADIUS security.</i></p>
Insufficient Resources	The total number of rejects due to a server resource problem.
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 may have occurred.

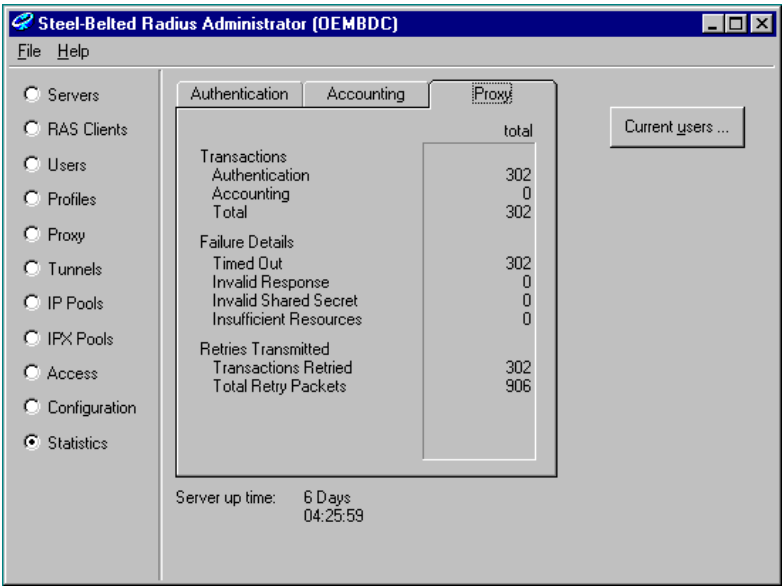


Figure 4-54: Statistics Dialog with Proxy Tab

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.
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.
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.

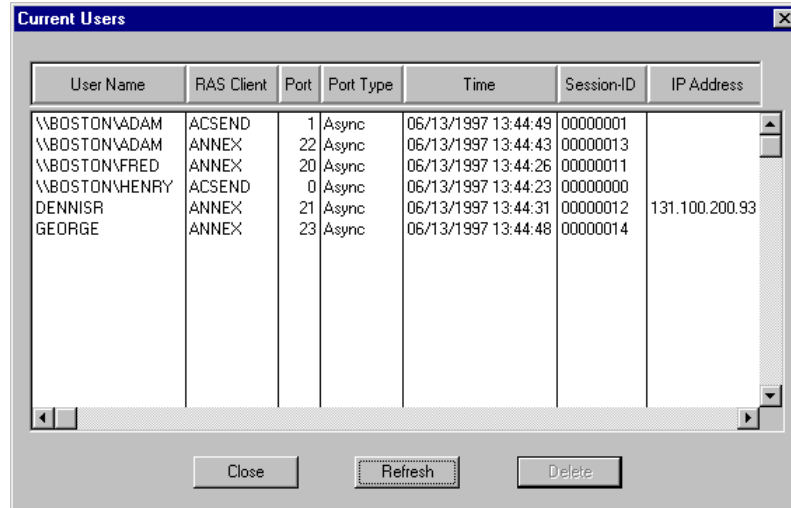
Current Users

Steel-Belted Radius tracks all users currently dialed in to a Remote Access Server.

The information about each connected user is based on accounting information received from the RAS. The Current User list will only be available if your RAS is configured to perform accounting.

Steel-Belted Radius maintains this information on disk. The information will not be lost if you unload and reload the server.

To display a report showing all Current Users, while viewing the Statistics dialog click **Current Users**.



User Name	RAS Client	Port	Port Type	Time	Session-ID	IP Address
\\BOSTON\\ADAM	ACSEND	1	Async	06/13/1997 13:44:49	00000001	
\\BOSTON\\ADAM	ANNEX	22	Async	06/13/1997 13:44:43	00000013	
\\BOSTON\\FRED	ANNEX	20	Async	06/13/1997 13:44:26	00000011	
\\BOSTON\\HENRY	ACSEND	0	Async	06/13/1997 13:44:23	00000000	
DENNISR	ANNEX	21	Async	06/13/1997 13:44:31	00000012	131.100.200.93
GEORGE	ANNEX	23	Async	06/13/1997 13:44:48	00000014	

Figure 4-55: Current Users Dialog

For every active dial-in session, a line is displayed in the report containing the following fields:

- ◆ **User Name** shows the name of the authenticated user. If the user is:
 - ◆ Native, the field will show the username only, in the form ***username***.
 - ◆ Non-native, the field will show the remote system name as well as the username, in the form ***\\systemname\\username***.
 - ◆ Associated with a specific tunnel, the field will show the tunnel name as well as the username, in the form ***\\tunnelname\\username***.
- ◆ **RAS Client** is the Remote Access Server (RAS) identification, which will either be the name of the RAS or its IP address.

- ♦ **Port** is the Remote Access Server (RAS) port number, which represents a unique port number on the RAS. To determine the actual physical port on the RAS, consult the RAS 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 generated by the RAS.
- ♦ **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 Column Size and Order

You can modify the order and size of the fields of the report 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.

Sorting

The report 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.

Getting Fresh Data

The Current Users report shows a snapshot of the current connected users taken when you open the dialog; to update the report with fresh information, click **Refresh**.

Deleting Entries

Normally, the system will take care of maintaining the correct information in the Current Users list based on accounting information received from the RAS. However, it is sometimes possible that a user who has logged off will still be indicated as active in the Current User list. This may occur as a result of communication failures between the RAS and Steel-Belted Radius, or it could occur if either the RAS 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 RAS, Steel-Belted Radius infers that the prior user must have disconnected and will remove the entry.

You can also manually correct the Current Users list by highlighting any entry and clicking **Delete**. In addition to removing the user from the list, the user's connection count (if it is being tracked) will decrease by one, and any pooled IP or IPX address that had been assigned to that user will be returned to the appropriate pool.

Server Up Time

Server Up Time displays the number of days, hours, minutes and seconds that Steel-Belted Radius has been active since it was last loaded.

Reporting

The **Report** command lets you assemble into a report any of the Steel-Belted Radius database information that is available through the Administration program's dialogs. For example, you can output to a report all the information you've set up about RAS Clients, Users, Proxies, and the like.

The **Report** command outputs the information in rich text format to a filename of your choice (normally **REPORT.RTF**), then opens that report using the word processor of your choice (normally **WORDPAD.EXE**). From the word processor, you can apply further formatting to the report, save it to an archive, or print it.

Setting Report Options

Before creating your first report, make sure that the settings for the output filename and the word processor with which you will view the report are correct:

- 1 Select the **File** command, then select **Settings**. The Settings dialog appears.

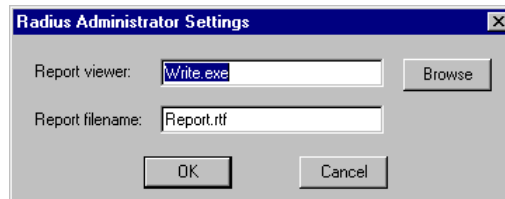


Figure 4-56: Settings Dialog

- 2 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).
- 3 When you are satisfied with the settings, click **OK**.

Creating a Report

To create a report:

- 1 Select the **File** command, then select **Report**. The Report Selections dialog appears.

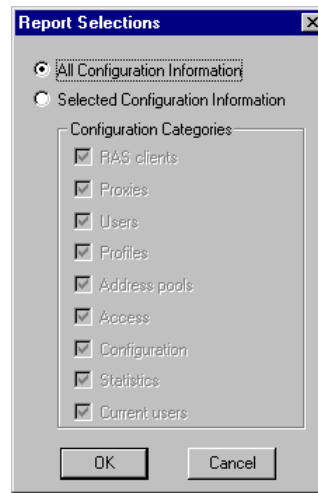


Figure 4-57: Report Selections Dialog

- 2 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.
- 3 Click **OK**. The report file will be created and will appear in your selected word processor.

Import/Export

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 you'd like 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 Select the **File** command, then select **Export**.

The Export dialog appears. Each tab in the dialog lists items of a particular category that you can export.

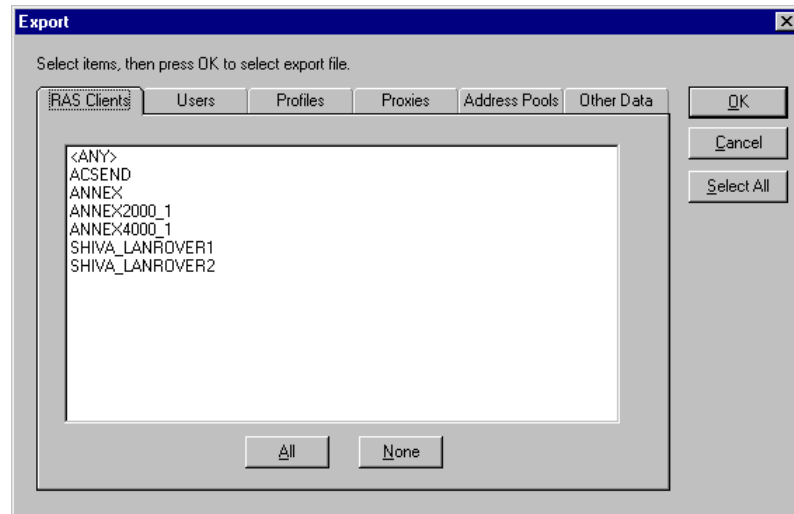


Figure 4-58: Export Dialog

- 2 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**.
- 3 Once you've selected all the items you want, click **OK**.
- 4 A file browsing dialog appears. Specify an export file and click **Save**.

Importing from a RADIUS Information File

To import from a RADIUS Information File into your Steel-Belted Radius database:

- 1 Select the **File** command, then select **Import**.
- 2 A file browsing dialog appears. Make sure the file type indicates **RADIUS Information File (*.rif)**. Select an import file and click **Open**.

The Import dialog appears, with each tab listing items of a particular category that are available for import from the selected file.

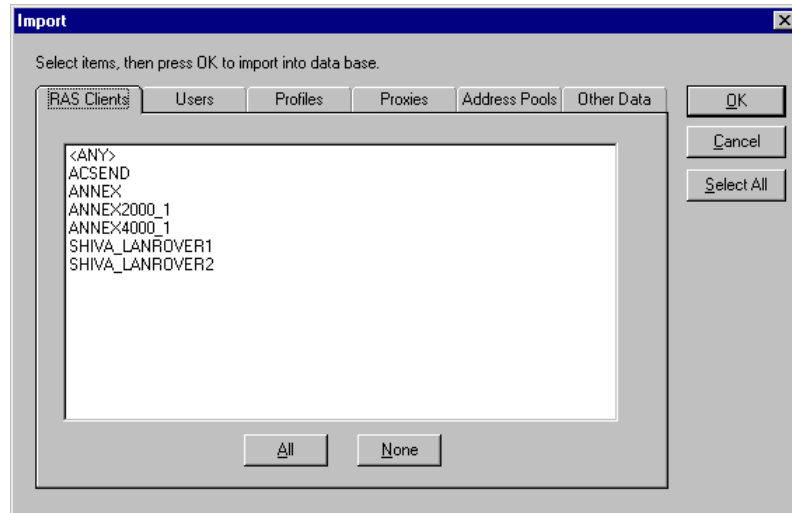


Figure 4-59: Import Dialog

- 3 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**.
- 4 Once you've selected all the items you want, click **OK**. The items you selected will be added to the Steel-Belted Radius database.

If an import item already exists, you will be given the opportunity to confirm that you'd like to replace the existing entry with the entry from the import file.

Importing from Other File Formats

There are many implementations of RADIUS currently deployed mostly on UNIX systems that are based on source code distributed by Livingston Enterprises and Ascend. These implementations store user 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 Select the **File** command, then select **Import**.
- 2 A file browsing dialog appears. Make sure the file type indicates **External User Information File**. Select an import file and click **Open**.

- 3 The Import Options dialog appears. Modify as desired:
- ♦ Set **File format** based on the RAS that the originating RADIUS server was meant to work with. This allows Steel-Belted Radius to correctly interpret attribute nomenclature of a particular vendor.
 - ♦ Check **Ignore attributes** if you'd like to import names and passwords only.
 - ♦ Select the **Profile** to be applied to each imported user entry, or leave the entry set to **<no profile>**. If you do select a profile, you'll probably also want to check **Ignore attributes**.
 - ♦ 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**. Once you've selected all the users you want, click **OK**.

The users you selected will be added to the Steel-Belted Radius database as Native users. If a user is already present in the database, you'll be given the opportunity to confirm that you'd like to replace the existing user with the new user entry from the import file.

Logging and Monitoring Features

5

Logging Features

Steel-Belted Radius provides logging and diagnostic features, including:

- ♦ Performance monitoring using the Windows NT Performance Monitor program
- ♦ An authentication log file, containing information for all authentication and administrative events
- ♦ An accounting log file, containing information for all start and stop accounting events.

Performance Monitor Graphing

The Steel-Belted Radius NT service has information which can be viewed with the Performance Monitor on Windows NT.

To view a graph of Steel-Belted Radius performance:

- 1 Start **PERFMON.EXE** (Windows NT Performance Monitor).
- 2 Click **Edit** then **Add to Chart**. Select Steel-Belted Radius from the list of Objects.
- 3 Select the counter you wish to graph. Select the Color, Scale, and any other options you want for the display. Click **Add**.

(For example, to see the Steel-Belted Radius transaction rate select **Auth Requests per sec.**)
- 4 When you are finished adding items to display, click **Done**. You can now see a graph of all your selected items.

*NOTE: You can view statistics for any Steel-Belted Radius server on your network. Simply select the computer name of another server. In addition, you can start multiple versions of **PERFMON.EXE** to view more than one Steel-Belted Radius server at one time.*

Authentication Log File

Each time a RADIUS authentication event occurs it is recorded in the authentication log file. Some examples of events include:

- ♦ Accepted authentications
- ♦ Rejected authentications
- ♦ Administrative actions
- ♦ Loading and unloading of RADIUS on the server

The log files are located in the RADIUS database directory area, and 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.

All log files are in ASCII format, with each line representing a RADIUS authentication event.

NOTE: The RADIUS log files will be kept for the number of days specified in the Administrator. After that time, older files will be deleted from the server in order to conserve disk space.

Authentication Log File Format

Each line of the authentication log file contains a line with the date and time, followed by event information. The authentication log is intended for viewing by the administrator, and can be opened while Steel-Belted Radius is running.

The following is a list of common authentication events:

- ♦ 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 ...

Accounting Log File

Each time a RADIUS accounting event occurs it is recorded in the accounting log file. Accounting events include:

- ♦ Start records, indicating the beginning of a connection.
- ♦ Stop records, indicating the ending of a connection.

The log files are located in the RADIUS database directory area, and 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.

All log files are in ASCII format, with each line representing a RADIUS accounting event.

NOTE: The RADIUS log files will be kept for the number of days specified using the Steel-Belted Radius Administrator program. After that time, older files will be deleted from the server in order to conserve disk space.

Accounting Log File Format

The accounting log file uses comma-delimited format, and is intended for import into a spreadsheet or database program.

NOTE: The accounting log file can be opened while Steel-Belted Radius is running.

First Line Headings

The first line of the accounting log file lists the names of all the attributes that have been enabled for logging. By default, this list consists of all the standard RADIUS accounting attributes, plus a complete set of additional attributes specific to each network access server product known to Steel-Belted Radius. This first line serves as a complete set of “column headings” for the remaining entries in the file.

The complete first line of an accounting log file might appear as follows. This example lists Steel-Belted Radius’s full set of standard RADIUS and vendor-specific attributes. Standard attributes are highlighted with boldface type.

```
"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", "Retrains-Requested", "Retrains-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 listed 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,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
```

Configuring the Accounting Log File

You can configure the accounting log file to enable or disable logging for various attributes. By default, all attributes are logged to the file. You can edit the **account.ini** initialization file to eliminate, add, or change the order of any vendor-specific attributes, as well as most of the standard RADIUS attributes. See the next chapter, “Dictionary and Initialization Files,” for details.

NOTE: The first five attributes in each log file entry (Date, Time, RAS-Client, Record-Type, and Full-Name) are always enabled, and cannot be re-ordered or deleted.

Standard Attributes

The standard attributes that can be written to the accounting log file provide the following information. The first 5 fields are generated by Steel-Belted Radius for convenience in reading 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, ON, or OFF
- ♦ **Full-Name** - the fully distinguished name of the user, based on the authentication from the RADIUS server

Next, you may see these standard RADIUS accounting attributes:

- ♦ **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
 - 15 - TACACS+ User
 - 16 - TACACS+ Prefix
 - 17 - TACACS+ Suffix
 - 100 - Tunnel User
 - 200 - External Database
 - (other) - Proxy
- ♦ **User-Name** - the name of the user as received by the RAS
- ♦ **NAS-Port** - the Remote Access Server port number
- ♦ **Acct-Status-Type** - a number that indicates the beginning or ending of user service:
 - 1 - Start
 - 2 - Stop
 - 7 - Accounting-On
 - 8 - Accounting-Off

- ♦ **Acct-Delay-Time** - indicates how many seconds the RAS Client has been trying to send this record; 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 bytes (characters) received by the port over the connection; only present in STOP records
- ♦ **Acct-Output-Octets** - number of bytes (characters) sent by the port over the connection; only present in STOP records
- ♦ **Acct-Session-Id** - unique Accounting identifier to make it easy to match START and STOP records in a log file
- ♦ **Acct-Authentic** - indicates how the user was authenticated, whether 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; only present in STOP records
- ♦ **Acct-Input-Packets** - number of packets received by the port over the connection; only present in STOP records
- ♦ **Acct-Output-Packets** - number of packets sent by the port over the connection; only present in STOP records
- ♦ **Acct-Termination-Cause** - a number that indicates how the session was terminated; only present 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** - gives the count of links which are known to have been in a given multi-link session at the time the accounting record is generated.

Following these standard attributes, vendor-specific RADIUS accounting attributes may appear in the file.

Initialization and Dictionary Files

6

Initialization Files

Initialization files are text files that run at startup time to configure Steel-Belted Radius behavior. This section lists the files and describes their syntax.

radius.ini File

The **radius.ini** initialization file contains information that controls the behavior of Steel-Belted Radius in a variety of ways. **radius.ini** is loaded at startup and resides in the Steel-Belted Radius server directory that you defined at installation time (usually **C:\RADIUS\Service**).

radius.ini is divided into sections. Section names are enclosed in square brackets, for example the **[CONFIGURATION]** section.

[CONFIGURATION] Section

The **[CONFIGURATION]** section of **radius.ini** contains parameters that control various aspects of Steel-Belted Radius's behavior. The following fields may be present:

Field	Meaning
Allow-Unmasked-Password	If set to Yes , it will be possible through the Administration program to view previously entered passwords (provided they are not strongly encrypted). If set to No (the default), it will be possible to unmask passwords as you enter them, but not to view passwords that have already been entered.
Allow-Unmasked-Secret	If set to Yes , it will be possible through the Administration program to view previously entered shared secrets. If set to No (the default), it will be possible to unmask shared secrets as you enter them, but not to view shared secrets that have already been entered.
Apply-Login-Limits	If set to Yes (the default), then the maximum number of concurrent connections for each user will be enforced, and connection attempts above the limit will be rejected. If set to No , then connections above the limit will be allowed, but an event will be noted in the authentication log file.

Field	Meaning
LogLevel	The logging level between 0 and 2 , where 0 is the default logging level, and 2 is most verbose. The log level specification can be used in lieu of specifying a -l2 on the load line for the service/console application.
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 RADIUS Server resides).
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 always written to the log file and can be a useful tool for trouble-shooting interoperability problems.

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** is loaded at startup and resides in the Steel-Belted Radius server directory that you defined at installation time (usually **C:\RADIUS\Service**).

vendor.ini contains one section only, the **[VENDOR-PRODUCT IDENTIFICATION]** section.

*NOTE: In previous releases of Steel-Belted Radius, this section was provided in the **radius.ini** file. There was no **vendor.ini** file in these releases.*

[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:

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 RAS client or when selecting a Vendor Specific attribute.
dictionary	<p>This required field specifies the dictionary file to use for this product. The dictionary file must be located in the Steel-Belted Radius server directory that you defined at installation time (usually C:\RADIUS\Service). You do not need to specify a extension on the dictionary name; Steel-Belted Radius automatically attaches an extension of .DCT to the dictionary names listed in this field.</p> <p>For additional details, see “Dictionary Files” below.</p>
send-class-attribute	If set to No , the Class attribute will not be sent to the RAS on Access-Accept. This is to accommodate RAS's that don't handle this attribute properly. The default is Yes .
ignore-acct-ss	If set to Yes , the digital signature of accounting packets based on the shared secret is ignored. This is to accommodate RAS's that don't properly sign accounting packages. The default is No .
ignore-ports	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. If set to No , then such an inference will be made and the previous user will be removed from the Active Users list. If set to Yes , then no such inference will be made and both users will be deemed active. The default is No .
discard-after	Used for inbound Proxy RADIUS servers which send username information in a parsable format. For example, if a Proxy RADIUS server sends usernames of the form username@company , then specifying @ will result in all text after the @ delimiter character being discarded for authentication purposes; the string username will be used.
discard-before	Same as discard-after, except the name is on the right of the delimiter character and discardable information is on the left.

account.ini File

The **account.ini** initialization file contains information that controls which RADIUS accounting attributes will be logged by Steel-Belted Radius. **account.ini** is loaded at startup and resides in the Steel-Belted Radius server directory that you defined at installation time (usually **C:\RADIUS\Service**).

account.ini contains one section only, the **[CONFIGURATION]** section.

*NOTE: In previous releases of Steel-Belted Radius, vendor-specific accounting attributes were enabled or disabled for accounting by using the **a** or **A** flag in each vendor-specific dictionary file. There was no **account.ini** file in these releases.*

[CONFIGURATION] Section

The **[CONFIGURATION]** section of **account.ini** lists all the attributes that will be logged in the Steel-Belted Radius accounting log file. When you first install Steel-Belted Radius, the **account.ini** file is set up so that all vendors' accounting attributes and all standard RADIUS attributes are listed.

You can configure what is logged to the accounting log file by rearranging the order of attributes in **account.ini** or by deleting or commenting out any attributes that are not of interest to you; for example, any vendor-specific attributes that do not apply to the equipment that you are using. This will allow you to design the content and column order of any spreadsheets that you plan to create based upon the accounting log file.

*NOTE: The first five attributes in each log file entry (Date, Time, RAS-Client, Record-Type, and Full-Name) are always enabled, and cannot be re-ordered or deleted. Therefore, these attributes do not appear in the **account.ini** file.*

The following fields may be present in the **[CONFIGURATION]** section of the **account.ini** file:

Field	Meaning
Attributes = <attribute1> \ <attribute2> \ <attribute3> # <attribute4> # <attribute5>	An exclusive list of attributes to be logged in the accounting log file. Only the attributes listed here will be logged, and they will be logged in the order listed. Any accounting attributes not listed will be available to the Steel-Belted Radius server, but they will not be written to the accounting log file.

A space, a \ backslash character, and a carriage return *must* appear between each name in the list of attributes. The \ character indicates that the list will continue on the next line. It also shows where each comma delimiter will appear in the accounting log file. With the last attribute, the list ends, so the last name in the list should not have a backslash after it.

The # character comments out a line, as long as the # is the first non-space character in the line and the # does not follow a \ character at the end of the previous line.

NOTE: If you want to comment out an attribute that appears in the middle of the list, move it down past the end of the list before applying the comment character to it.

We strongly suggest that, before you make any changes to the **account.ini** file, you make a backup copy of it.

tacplus.ini File

The **tacplus.ini** initialization file provides the configuration information that enables the Steel-Belted Radius server to talk to a TACACS+ server. **tacplus.ini** is loaded at startup and resides in the Steel-Belted Radius server directory that you defined at installation time (usually **C:\RADIUS\Service**).

You must edit **tacplus.ini** to identify the shared secret and host machine that you use for TACACS+. If you edit **tacplus.ini** after Steel-Belted Radius has been started, then you must stop and restart Steel-Belted Radius before your changes will take effect.

tacplus.ini contains one section only, the **[ServerInfo]** section.

[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:

Field	Meaning
SharedSecret=	The shared secret between the TACACS+ server and Steel-Belted Radius. For example: TACPLUS123
TargetHost=	The IP address of the TACACS+ server. For example: 197.43.160.101

services File

Steel-Belted Radius reads the **WINNT\system32\drivers\etc\services** file at startup. Among the items of information in the **services** file are the port assignments for RADIUS authentication and accounting services.

The standard ports to use for RADIUS authentication and accounting were **1645** and **1646** (respectively) when the standard was first written. Then it emerged that these ports had been assigned to another standard. The RADIUS standards group responded by changing the port assignments to **1812** and **1813**, but most organizations still use the old assignments.

The Steel-Belted Radius server uses the following default ports:

- ♦ **1645** for RADIUS authentication
- ♦ **1646** for RADIUS accounting

The Steel-Belted Radius server can be configured to use *any* UDP port for authentication and accounting; for example, ports **1812** and **1813**. You can configure these port assignments 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 will use the default values (**1645** and **1646**).

You can determine the ports that Steel-Belted Radius is using at any time by examining the Authentication log files **yyyymmdd.log** or the Accounting log files **yyyymmdd.act** for that time period.

Dictionary Files

RADIUS authentication packets (request, accept and reject) and accounting packets (start and stop) consist of a common packet header followed by a collection of one or more attributes. Each attribute in a RADIUS packet is accompanied by an attribute value. The dictionary files provided as part of Steel-Belted Radius provide the necessary information for:

- ♦ Decoding an incoming authentication request or accounting packet
- ♦ Encoding an outgoing authentication accept or reject

Each product defined in the **vendor.ini** initialization file includes a dictionary file to be used when dealing with the RADIUS clients of that product type.

The remainder of this section describes the syntax supported in the dictionary files and the type of dictionary files supported by Steel-Belted Radius.

NOTE: Network administrators will not usually be required to modify the dictionary files shipped with Steel-Belted Radius. The product-specific files shipped with Steel-Belted Radius already reflect specific vendors' implementations of RADIUS clients. The information in this section is provided for reference purposes only.

Basic Rules for Dictionary Files

Dictionary files must be placed in the Steel-Belted Radius server directory that you defined at installation time (usually **C:\RADIUS\Service**). During initialization, Steel-Belted Radius scans for all files with an extension of **.dct** in the server directory and attempts to process any such files as dictionary files.

Dictionary files are processed in a record oriented manner; no mechanism is included for continuing a statement across multiple records. Records must begin with one of the following:

String	Meaning
ATTRIBUTE	Defines a new attribute
VALUE	Defines a "named integer value" for an attribute
MACRO	Defines a macro used to simplify repetitive definitions
OPTION	Defines options beyond the scope of attribute definitions
@	Include records (described below)
#	Comment lines (have no real effect)
	Blank lines are ignored

Attribute, Value, Macro, Option and Include records are described in more detail below.

The ATTRIBUTE Record

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
<attrib_name>	Name of the attribute (up to 31 characters with no embedded blanks)
<attrib_id>	Integer in the range 0 to 255 identifying the attribute's encoded identifier
<syntax_type>	Syntax type of the attribute (see the "Syntax Type Identifiers" section below)
<flags>	Defines whether an attribute appears in the check list, the reply list (or both), whether it is multi-valued and whether it is orderable (see the "Flag Characters" section below)

The following example illustrates a typical attribute record:

```
ATTRIBUTE    Framed-IP-Netmask      9      ipaddr    Cr
```

This record specifies that an attribute named Framed-IP-Netmask is supported, that its encoded identifier is 9, that it is to adhere to the syntax of an IP address, that it can appear multiple times in a check-list and at most one time in a reply-list for user or profile records in the database.

Attribute Name and Identifier

No two attribute records in a 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)

In addition to the standard **<syntax_type>** identifiers listed above, the dictionary can accommodate compound syntax types for use in defining vendor-specific attributes. In lieu 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
vid=nnn	RAS vendor's SMI Network Management Private Enterprise code (assigned by ISO) in decimal form
typeN=nnn	Type field for vendor-specific attribute as defined in the RADIUS specification; N specifies the length of the field (in bytes), nnn specifies the decimal value of the field
lenN=nnn	Length field for vendor-specific attribute as defined in the RADIUS specification; N specifies the length of the field (in bytes), nnn 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 nnn to obtain the actual length)
data=<syntax_type>	The actual data to be included in the attribute; the syntax can be any of the standard syntax types

An example of a vendor-specific attribute definition follows:

```
ATTRIBUTE cisco-xxx 26 [vid=9 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
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 reply-list
R	Attribute can appear multiple times within a user or profile reply-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 database (this only makes sense for multi-valued attributes)

The VALUE Record

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
<attrib_name>	Name of the attribute (up to 31 characters with no embedded blanks)
<value_name>	Name of the attribute value (up to 31 characters with no embedded blanks)
<integer_value>	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 will allow you to choose from a list of attribute values including **PPP** and **SLIP**.

The MACRO Record

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
<macro_name>	Name of the macro
<macro_vars>	One or more comma-delimited macro variable names
<subst_string>	String into which macro variables are to be substituted; any sequence of characters conforming to the format %x% for which a macro variable called x has been defined will 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
```

Using the macro preprocessor built into the Steel-Belted Radius dictionary processing, the records in the example above would be translated to the following records before being further 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
```

The OPTION Record

Option records are used to specify vendor-specific options that fall outside the scope of formatting individual attributes. The option record must conform to the following format:

```
OPTION <option_name>=<option_value>
```

where the parameters have meaning as follows:

Parameter	Meaning
<option_name>	Name of the option; only currently supported option is bundle-vendor-specific-attributes
<option_value>	Setting for the specified option; currently supported values (for bundle-vendor-specific-attributes) are yes and no

The **bundle-vendor-specific-attributes** option controls whether or not multiple vendor-specific attributes are bundled into a single attribute prior to being inserted into a RADIUS response.

The Include Record

Records that begin with the @ character are treated as special include records. The string that immediately follows the name identifies the name of a dictionary file whose contents are to be included in place of the include record.

Include records are only honored one level deep. If, for example, file **vendora.dct** specifies an inclusion of file **radbase.dct** which, in turn, includes **radacct.dct**, **vendora.dct** will be considered to include all records in **radbase.dct**, but not those in **radacct.dct**.

The Master Dictionary File

The master dictionary file is a special dictionary file that is only required by the Steel-Belted Radius Administrator program. In order to allow attributes from multiple dictionaries to be combined into a single dictionary so that users can be configured with attributes for multiple vendor's products, Steel-Belted Radius processes a special dictionary called **dictiona.dcm** and makes its contents available to the Administrator program

The master dictionary typically consists of some 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.

The one limitation of standard dictionary files that is waived for the master dictionary is that the **<attrib_id>** of all the attribute records must be unique. 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 are special dictionary files that are only used when importing user data from a text file supported by Unix implementations of RADIUS. In order to map the names of attributes commonly used in these implementations to those that are in use in Steel-Belted Radius, import dictionaries are provided.

The Steel-Belted Radius server reads all files with an extension of DCI in the Steel-Belted Radius server directory that you defined at installation time (usually **C:\RADIUS\Service**) and attempts to process any such files as import dictionary files. The expected format of records in the import dictionaries is identical to that of records in standard dictionaries.

When the Administration 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.

Using Steel-Belted Radius with SQL

7

Using Steel-Belted Radius with SQL

As described in Chapter 4 “RADIUS Concepts,” Steel-Belted Radius can use an external SQL database to store and retrieve RADIUS authentication or RADIUS accounting information. You must configure both Steel-Belted Radius and the SQL database to support this feature.

The exact configuration procedure must be tailored to the database that you use. However, all procedures must give the following results:

- ♦ The SQL server will be configured to be listening for client requests. Note that for SQL purposes, the Steel-Belted Radius server will be a client of the SQL server.
- ♦ The Steel-Belted Radius server will “know” about the remote SQL server. That is, it will know the machine where the SQL server software runs, and it will know the protocol and port used in communicating with that machine.
- ♦ The required transport will be in place between SQL client and server.

To assist you in configuring Steel-Belted Radius for use with SQL, an instruction file is provided for each type of SQL database that Steel-Belted Radius supports. Look for these files in the Steel-Belted Radius server directory that you defined at installation time (usually **C:\RADIUS\Service**). The naming convention for these files is **README_<DATABASE TYPE>.TXT**; for example **README_INFORMIX.TXT**.

Authenticating to an SQL Database

Steel-Belted Radius offers a plug-in **SQL Authentication** module that allows you to use a backend SQL database as a repository of username and password information for authentication.

Key features 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.

- ♦ 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.

CAUTION: Multi-vendor spoken here. While Steel-Belted Radius does its best to provide uniformity in the operation of databases from different vendors, there will be differences, particularly in the way SQL statements are interpreted. The capabilities of the SQL Authentication module are dependent upon the capabilities of the underlying databases and their clients; things that work with one database may not work with another.

Database Fields

The database must at a minimum contain the following information:

- ♦ **Username.** This field is used to match the login name entered by the user; it should be indexed, for rapid lookup.
- ♦ **Password.** This field is returned and matched against the password entered by the user.

In addition, the following fields may be present:

- ♦ **Profile.** This field, if present, is used to select a profile that specifies the check-list and return-list attributes to be used for this transaction. Profiles are configured using the administration program's Profiles dialog. If the profile retrieved from the database does not match one of the configured profiles, the user is rejected.
- ♦ **FullName.** This field, if present, specifies the complete name of the user, for accounting purposes.

The Header File

To configure SQL Authentication, you must edit the header file, **radsql.aut**, located in the same directory that contains the Steel-Belted Radius server executable. This is normally **C:\RADIUS\Service**.

Information is read out of the header file at startup. If any changes are made to the header file, the service must be restarted before the changes will take effect.

The [Bootstrap] section of the header file contains basic information for starting up the SQL Authentication module:

```
[Bootstrap]
LibraryName=radsq1_auth_ora.so
Enable=1
InitializationString=SQL
```

The LibraryName entry must contain the name of the module. Separate modules are provided for accessing databases from different vendors. See the reference section, below, for information about the module name that corresponds to your database.

The Enable entry must contain a **1** to enable the module, **0** to disable it. If disabled, the authentication method is unavailable and will not appear in the Configuration dialog's list of authentication methods.

The InitializationString must be set to the name to assign to this authentication method; this name appears in the Configuration dialog, allowing you to activate, deactivate, and order this method with respect to other methods. The method name is initially set to "SQL"; you may modify it if you like.

Other header file options are described below. A reference listing of all header file options appears at the end of this section. Most of these options may be left at their original settings; however, you will need to modify certain options in order to accommodate your own database.

Using Multiple SQL Databases

If you like, you can configure Steel-Belted Radius to authenticate users against more than one SQL database, and you can set the order in which databases are tried in the Configuration dialog.

To add an additional database, create a new header file with extension **.aut** in the Steel-Belted Radius server directory (usually **C:\RADIUS\Service**). 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 **radsq1.aut**. 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 administrator interface.

Connecting to the 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 will need to 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. Otherwise, at startup and each time a reconnect is required a pop-up dialog will appear into which the password must be typed prior to 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 backoff strategy in determining when to attempt a new connection. 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.

Overlapped Execution of SQL Statements

SQL Authentication can be configured with a maximum number of simultaneous executions of any SQL statement, using the MaxConcurrent entry.

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.

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.

SQL Statement Construction

The authentication transaction is based on a SQL query which 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 will need to configure the SQL entry of the header file with a query appropriate to your database. The query you enter must be a SELECT statement, and will contain additional syntax elements that 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 will 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	Type	Meaning
%name	string	the name of the user attempting to log in
%password	string	the password supplied

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 will always be 's' for string. For example, if the user's name is stored in the database as a string of up to 63 bytes, you'd 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.

The SQL Authentication module expects up to three columns to be returned by the query, in order:

Column	Meaning
password	the user's password, in clear text
profile	the name of the profile to associate with the user
full name	the full name of the user

The SQL Authentication module uses the order of the columns as they appear in the SQL statement to determine the type of information each column contains. The names of these columns may be anything at all, as long as the columns are specified in the correct order. All the columns need not be present as long as the order is preserved; for example, full name may be left out. However, if profile is left out, the software will mistake the full name for the profile unless a placeholder is used. This is easy in SQL; in the profile column position, just insert an empty string (enclosed in single quotes) with the AS keyword and a dummy field name:

```
SELECT password, '' AS whatever, fullname FROM  
usertable WHERE username = %name/63s
```

Including the Password in the Query

Normally, the only parameter you'd include in the SQL statement is %name. You might be wondering why %password is also available.

The only real purpose for this is 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 arrange to authenticate them correctly with a query such as this:

```
SELECT password, profile, fullname FROM usertable
WHERE username = %name/63s and
password = %password/63s
```

Note that you still have to return the password as the first column of the result in order to perform the authentication.

Header File Reference

The header file used to configure the SQL Authentication module must have extension **.aut**. 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.

[Bootstrap] Section

The [Bootstrap] section specifies information that Steel-Belted Radius uses to load and start the SQL Authentication module.

[Bootstrap] Entry	Meaning for SQL Authentication
LibraryName	This entry must be set to the module name. You must select the correct module name based on the type of database you have. A separate version of the SQL Authentication module is included for each supported database: radsql_auth_ora.so (for Oracle) radsql_auth_inf.so (for Informix)
Enable	This entry is set to 0 to disable, 1 to enable the SQL Authentication module.

[Bootstrap] Entry	Meaning for SQL Authentication
InitializationString	<p>This entry is used to specify the name of the authentication method to appear in the list of authentication methods in the Administrator's Configuration dialog.</p> <p>In the original header file, the method name is set to "SQL". You may alter this name if you wish.</p> <p>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.</p>

[Settings] Section

The [Settings] section defines parameters that control aspects of the database connection.

[Settings] Entry	Meaning for SQL Authentication
Connect	<p>The Connect entry specifies the string that must be passed to the database client engine to establish a connection to the database. This string will have, or refer 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 will vary. See the configuration instruction file for the database you are using. You will find it in the same directory that contains the Steel-Belted Radius server executable (usually C:\RADIUS\Service).</p>
ConnectTimeout	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.
WaitReconnect	WaitReconnect specifies the number of seconds to wait after a failure of the database connection before trying to connect again.

[Settings] Entry	Meaning for SQL Authentication
MaxWaitReconnect	<p>MaxWaitReconnect 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>
SQL	<p>The SQL entry contains the SQL statement used to access the password information in the database.</p> <p>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.</p> <p>Example:</p> <pre>SQL=SELECT password, profile, fullname \ FROM usertable \ WHERE username = %name/63s</pre>
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>
MaxConcurrent	<p>MaxConcurrent specifies the maximum number of instances of a single SQL statement that may be executing at one time.</p>
ConcurrentTimeout	<p>Since there may only be up to MaxConcurrent SQL statements executing at one time, as new requests arise they must be queued, waiting for other statements to complete. ConcurrentTimeout specifies the number of seconds a request may wait for execution before it is discarded.</p>
QueryTimeout	<p>QueryTimeout specifies the number of seconds to wait for a response to a query before timing out. This value is passed to the client database engine, which may or may not implement the feature.</p>
UpperCaseName	<p>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.</p>

SQL Accounting

Steel-Belted Radius offers a plug-in **SQL Accounting** module that allows you to log accounting transactions to a backend SQL database.

Key features 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.

CAUTION: Multi-vendor spoken here. While Steel-Belted Radius does its best to provide uniformity in the operation of databases from different vendors, there will be differences, particularly in the way SQL statements are interpreted. The capabilities of the SQL Authentication module are dependent upon the capabilities of the underlying databases and their clients; things that work with one database may not work with another.

The Header File

To configure SQL Accounting, you must edit the header file, **radsql.acc**, located in the same directory that contains the Steel-Belted Radius server executable (usually **C:\RADIUS\Service**).

Information is read out of the header file at startup. If any changes are made to the header file, the service must be restarted before the changes will take effect.

The [Bootstrap] section of the header file contains basic information for starting up the SQL Accounting module:

```
[Bootstrap]
LibraryName=radsql_acct_ora.so
Enable=1
InitializationString=
```

The LibraryName entry must contain the name of the module. Separate modules are provided for accessing databases from different vendors. See the reference section, below, for information about the module name that corresponds to your database.

The Enable 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**.

The InitializationString is unused.

Other header file options are described below. A reference listing of all header file options appears at the end of this section. Most of these options may be left at their original settings; however, you will need to modify certain options in order to accommodate your own database.

Using Multiple SQL Databases

If you like, 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 Steel-Belted Radius server directory (usually **C:\RADIUS\Service**). 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.

When creating the new file, start by copying the original **radsql.acc**, then make whatever modifications are necessary.

Connecting to the 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 will need to 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. Otherwise, at startup and each time a reconnect is required a pop-up dialog will appear into which the password must be typed prior to 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 backoff strategy in determining when to attempt a new connection. 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.

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 of interest to one of these SQL statements. The different statements may update different tables in the database, but they all share the single database connection.

NOTE: If you want to update multiple databases, you must create multiple header files to run multiple instances of the SQL Accounting module.

Overlapped Execution of SQL Statements

SQL Accounting can be configured with a maximum number of simultaneous executions of any SQL statement, using the MaxConcurrent entry.

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**. The optimum value is a matter of experiment.

Note that 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. So another factor that determines how MaxConcurrent should be set might be the database license.

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 will need to configure one or more SQL entries of the header file with a statement appropriate to your database. Each SQL statement will normally be an INSERT INTO statement, and will 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 will 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.

Example:

```
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)
```

As shown in the example above, each parameter is prefixed with either an at sign (@) or a percent sign (%):

- ♦ @ indicates a RADIUS accounting attribute. The attribute name must match an attribute listed in the **account.ini** initialization file.
- ♦ % indicates the item associated with the request that is not an attribute. The following items are available:

Item	Type	Meaning
%TransactionTime	time	The date/time that the event occurred that is the subject of the request
%Time	time	The date/time right now when the request is being processed. (This will be a later than %TransactionTime if the request is a retry.)
%Type	String	The type of request, based on Acct-Status-Type. The [TypeNames] section maps each Acct-Status-Type to a string
%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 directly 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 type.

The following types are available:

Specifier	Meaning
/<length>s	string; if length not given, defaults to 256
/n	32-bit integer
/n8	8-bit integer
/n16	16-bit integer
/n32	32-bit integer (same as /n)
/t	timestamp

If a format specifier is not present; Steel-Belted Radius will automatically default to an appropriate specifier based on the actual parameter type. For example, @Acct-Input-Octets is a number, and will default 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 will default 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 will be equivalent; if not, SBR will perform reasonable conversions. The following table lists the internal formats and their compatible database formats:

Internal Format	Database Format
number	/n, /n8, /n16, /n32, /<length>s
string	/<length>s
time (seconds since 1/1/70)	/t, /n, /n32, /<length>s
IP address	/n, /n32, /<length>s

Header File Reference

The header file used to configure the SQL Accounting module must have 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.

[Bootstrap] Section

The [Bootstrap] section specifies information that Steel-Belted Radius uses to load and start the SQL Accounting module.

[Bootstrap] Entry	Meaning for SQL Accounting
LibraryName	This entry must be set to the module name. You must select the correct module name based on the type of database you have. A separate version of the SQL Accounting module is included for each supported database: radsql_acct_ora.so (for Oracle) radsql_acct_inf.so (for Informix)

[Bootstrap] Entry	Meaning for SQL Accounting
Enable	This entry is set to 0 to disable, 1 to enable the SQL Accounting module.
InitializationString	This entry is unused.

[Settings] Section

The [Settings] section defines parameters that control aspects of the database connection.

[Settings] Entry	Meaning for SQL Accounting
Connect	<p>The Connect entry specifies the string that must be passed to the database client engine to establish a connection to the database. This string will have, or refer 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 will vary. See the configuration instruction file for the database you are using. You will find it in the same directory that contains the Steel-Belted Radius server executable (usually C:\RADIUS\Service).</p>
ConnectTimeout	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.
WaitReconnect	WaitReconnect specifies the number of seconds to wait after a failure of the database connection before trying to connect again.
MaxWaitReconnect	<p>MaxWaitReconnect 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>

[Settings] Entry	Meaning for SQL Accounting
ParameterMarker	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. The parameter marker is ignored by the Solaris Oracle authentication method which uses ':1', ':2', and so forth.
UTC	This entry should be set to 0 to show time information in local time, or 1 to show time information in coordinated universal time.
MaxConcurrent	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/<statement>] section for that statement.
ConcurrentTimeout	Since there may only be up to MaxConcurrent SQL statements executing at one time, as new requests arise they must be queued, waiting for other statements to complete. ConcurrentTimeout specifies the number of seconds a request may wait for execution before it is discarded. ConcurrentTimeout may be overridden for any particular statement in the [Type/<statement>] section for that statement.
QueryTimeout	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/<statement>] section for that statement.
UpperCaseName	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.

[Type] Section

The [Type] section maps the type of accounting request (Acct-Status-Type attribute) to a particular SQL statement.

Each [Type] entry maps an Acct-Status-Type value to a statement name which 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.

The following Acct-Status-Type values have been defined in the RADIUS RFCs.

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

Other Acct-Status-Type values have been defined by NAS vendors for use with their equipment. These vendor-specific values may also be specified in the [Type] section.

For example, to perform separate accounting updates for NAS and user activity, you might have the following:

```
[ Type ]
1=user
2=user
3=user
7=nas
8=nas
[ Type/user ]
...
[ Type/nas ]
...
```

[Type/<statement>] Section

Each [Type/<statement>] section governs a particular SQL statement that executes against the database. The [Type] section maps each Acct-Status-Type to a particular [Type/<statement>] section, as described above.

[Type/<statement>] Entry	Meaning for SQL Accounting
SQL	<p>The SQL entry contains the SQL statement used to update the database with accounting information.</p> <p>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.</p> <p>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>
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>

[TypeNames] Section

The [TypeNames] section maps the type of accounting request (Acct-Status-Type attribute) to a string. When a request is received for that Acct-Status-Type, the %Type parameter will contain the given string.

If no string is given for a particular type, %Type will be set to the numeric value of the Acct-Status-Type attribute, formatted as a string.

Index

[

- [Bootstrap]
 - radsq1.acc file, 145
 - radsq1.aut file, 136
- [Configuration]
 - account.ini file, 119
 - radius.ini file, 116
- [ServerInfo]
 - tacplus.ini file, 120
- [Settings]
 - radsq1.acc file, 146
 - radsq1.aut file, 137
- [Type/<statement>]
 - radsq1.acc file, 148
- [Type]
 - radsq1.acc file, 147
- [TypeNames]
 - radsq1.acc file, 149
- [Vendor-Product Identification]
 - vendor.ini file, 117

A

- Access Dialog, 87
- account.ini file, 119
 - [Configuration], 119
 - and SQL accounting, 143
- Accounting
 - configuring, 119
 - log file format, 110
 - port assignments for, 121
 - statistics, 95
 - using an SQL database, 139
- Adding a Group
 - domain group, 55
 - host group, 55

- Adding a User
 - domain user, 55
 - host user, 55
 - native, 54
 - SecurID, 56
 - TACACS+, 58
- Attribute Exchange, 25
 - dictionary files, 26, 121, 128
 - types of attribute, 63
- ATTRIBUTE record
 - dictionary files, 123
- Authentication
 - configuring methods for, 91
 - log file format, 109
 - port assignments for, 121
 - statistics, 93
 - using an SQL database, 130
- Authentication Types, 20
 - against a local database, 22
 - Domain Group, 22, 55
 - Domain User, 22, 55
 - external, 25
 - Host Group, 23, 55
 - Host User, 22, 55
 - Native User, 22, 54
 - proxy, 25, 30, 77
 - SecurID User, 24, 56
 - TACACS+ User, 25, 58

C

- Challenge Handshake Authentication Protocol, 21
- CHAP, 21, 61, 106
- Check-list Attributes
 - adding, 66
 - changing, 68
 - default values, 65
 - echo property, 64

- inherited from a profile, 65, 69
- multi-valued attributes, 64
- orderable attributes, 64
- removing, 68
- reordering, 68
- Concurrent Connections
 - tunnels, 41, 79
 - users, 40, 62
- Configuration Dialog, 89

D

- Databases
 - external authentication, 25
- dictiona.dcm file, 128
- Dictionaries, 19, 103, 128
- Dictionary files, 8, 26, 121, 128
 - ATTRIBUTE record, 123
 - basic rules, 122
 - Include record, 127
 - MACRO record, 126
 - OPTION record, 127
 - VALUE record, 125
- DNIS, 36
- Domain Authentication, 23
- Domain group
 - adding, 55
 - authentication of, 22
- Domain user
 - adding, 55
 - authentication of, 22

E

- Exporting, 104
- External Databases, 31
 - configuring Steel-Belted Radius for use with, 32

F

- files
 - account.ini, 119, 143
 - dictiona.dcm, 128
 - dictionary, 121

- radacct.dct, 127
- radadnt.exe, 44
- radbase.dct, 127
- radius.ini, 116, 122
- radsql.acc, 139, 145
- radsql.aut, 131, 136
- radsql_acct_inf.so, 145
- radsql_acct_ora.so, 145
- radsql_auth_inf.so, 136
- radsql_auth_ora.so, 136
- RAS product help file, 15
- sdconf.rec, 15, 24
- services, 121
- tacplus.ini, 15, 25, 120
- vendor.ini, 117

H

- Host Authentication, 24
- Host group
 - adding, 55
 - authentication of, 23
- Host user
 - adding, 55
 - authentication of, 22

I

- Import/Export, 103
- Importing, 128
 - other file formats, 105
 - RADIUS information file, 104
- Include record
 - dictionary files, 127
- INSERT statement, SQL, 149
- Installation
 - configuration, 14
 - overview, 12
 - updating a previous installation, 13
 - upgrading from a 30-day trial, 14
- IP address
 - setting for RAS client, 48
- IP Address Pool, 49

IP Address Pools

- adding, 81
- editing, 82
- removing, 82

IP Pools Dialog, 80

IPX Address Pools

- adding, 85
- editing, 85
- removing, 86

IPX Pools Dialog, 84

L

License

- 30-day, upgrading from, 14

Licensing, 8

Loading and Unloading the Radius Service, 14

Log file

- accounting format, 110
- authentication format, 109
- days to keep, 92
- how to find port assignments in, 121

Logging, 108

- accounting log file, 92, 110
- authentication log file, 108

M

MACRO record

- dictionary files, 126

Make/model

- setting for RAS client, 48

Microsoft Challenge Handshake Authentication

- Protocol, 21

MS-CHAP, 21

N

Native User

- adding, 54
- authentication of, 22

O

OPTION record

dictionary files, 127

P

PAP, 20, 21, 49, 61, 62, 106

Pass-through authentication

- against Microsoft Networking, 22
- SecurID, 24
- TACACS+, 25

Password Authentication Protocol, 20, 49

Performance Monitor Graphing, 108

ports, UDP

- configuring, 121

PPP, 126

Profile, 62, 69

- adding, 71
- editing, 72
- removing, 72

Profiles Dialog, 70

Proxy Dialog, 73

Proxy RADIUS, 27

- adding a new target server, 74
- adding a wildcard RAS client, 51
- adding roaming proxy support, 77
- editing Proxy settings, 75
- proxy accounting, 28, 76, 77
- proxy authentication, 25, 30, 77
- proxy forwarding, 28
- proxy statistics, 97
- removing a target server, 78
- roaming, 28, 29
- Steel-Belted Radius as target server, 29
- using DNS, 28

R

radacct.dct file, 127

radadnt.exe file, 44

radbase.dct file, 127

RADIUS

- accounting, 27
- authentication, 12, 19
- concepts, 18

- definition, 6
- dictionaries, 8, 19, 103, 128
- Steel-Belted Radius overview, 7
- radius.ini file, 116, 122
 - [Configuration], 116
- radsql.acc file, 139, 145
 - [Bootstrap], 145
 - [Settings], 146
 - [Type/<statement>], 148
 - [Type], 147
 - [TypeNames], 149
- radsql.aut file, 131, 136
 - [Bootstrap], 136
 - [Settings], 137
- radsql_acct_inf.so file, 145
- radsql_acct_ora.so file, 145
- radsql_auth_inf.so file, 136
- radsql_auth_ora.so file, 136
- RAS client
 - adding a new, 47
 - editing settings, 48
 - IP address, 48
 - make/model, 48
 - removing from list, 51
- RAS Clients Dialog, 46
- RAS product help file, 15
- Reject Messages, 90, 91
- Removing a User, 69
- Reporting, 101
- Return-list Attributes
 - adding, 67
 - changing, 68
 - default values, 65
 - echo property, 64
 - inherited from a profile, 65, 69
 - multi-valued attributes, 64
 - orderable attributes, 64
 - removing, 68
 - reordering, 68
- Roaming proxy, 28, 29, 77

S

- sdconf.rec file, 15, 24
- SecurID User
 - adding, 56
 - authentication of, 24
- SELECT statement, SQL, 138
- Servers Dialog, 45
- services file, 121
- Shared Secret, 49, 75, 77, 96, 98
- SLIP, 126
- SQL database
 - concurrent SQL statements, 133, 142
 - configuring for accounting, 139
 - configuring for authentication, 131
 - connecting to, 133, 140
 - for RADIUS accounting, 139
 - for RADIUS authentication, 130
 - INSERT statement, 149
 - SELECT statement, 138
 - supported versions, 8
 - using more than one, 132, 140
- Statistics
 - accounting, 95
 - authentication, 93
 - current users, 39, 98, 99
 - graphing, 108
 - proxy RADIUS, 97
- Statistics Dialog, 93
- System Requirements, 8

T

- TACACS+ User
 - adding, 58
 - authentication of, 25
- tacplus.ini file, 15, 25, 120
 - [ServerInfo], 120
- Technical Support, 9
- Tunnels, 32
 - adding, 79
 - concurrent connections, 41, 79
 - configuring Steel-Belted Radius to support, 34

- name parsing, 91
- removing, 80
- Tunnels Dialog, 78

U

- UDP ports
 - configuring, 121
- Users

- concurrent connections, 40, 62
- Users Dialog, 52

V

- VALUE record
 - dictionary files, 125
- vendor.ini file, 117
 - [Vendor-Product Identification], 117