



**Juniper Networks
Steel-Belted Radius**

Mobile IP Module Administration Guide

Release 5.3

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

Part Number: SPR-RZ-53MIMMN

Copyright © 2005–2006 Juniper Networks, Inc. All rights reserved. Printed in USA.

Steel-Belted Radius, Juniper Networks, and the Juniper Networks logo are registered trademarks of Juniper Networks, Inc. in the United States and other countries. CORBA (Common Object Request Broker Architecture) is a registered trademark of the Object Management Group (OMG). Raima, Raima Database Manager and Raima Object Manager are trademarks of Birdstep Technology. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners. All specifications are subject to change without notice.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

M071006

Contents

About This Guide

Who Should Read This Guide.....	ix
Before You Begin	ix
Documentation Conventions	x
For More Information	xi

Chapter 1

About 3GPP2 and the Mobile IP Module

Overview.....	1
IP Address Management	2
Simple IP	2
Mobile IP.....	2
IP Address Durability in Roaming Scenarios	3
Support for Special Accounting Requests.....	3

Chapter 2

3GPP2 Mobile IP Module Configuration

Product Registration.....	5
Simple IP Configuration.....	5
Simple IP Request Types	5
SIP User Authentication Requests	6
Mobile IP Configuration	6
Mobile IP Request Types.....	6
Determining Mobile IP Request Types	7
FA User Authentication Requests	11
Home Agent (HA) Assignment Process	12
Home Agent (HA) Assignment Summary.....	13
Home Address Assignment Process.....	14
Home Address Assignment Summary	15
IPsec Security Attributes Needed for FA/HA Communications.....	16
Proxy Realm Based Configuration.....	16
HA Key Distribution Requests	17
MN-HA Shared Key Distribution Requests	18
HA User Authentication Requests	19
IP Address Assignments from Filters.....	19

Attributes and Filters.....	20
Mobile IP Module Configuration	20
[Settings] Section.....	21
[Authorize-Only-Requests] Section	22
[FA-User-Auth-Requests] Section	23
[HA-Key-Distribution-Requests] Section.....	23
[MN-HA-Shared-Key-Requests] Section.....	25
[HA-User-Auth-Requests] Section.....	25
[SIP-User-Auth-Requests] Section.....	26
[Other-Requests] Section.....	26
[Attributes] Section.....	27
[HAs] Section	27
[FA-User-Auth-Requests/name] Sections	27

Chapter 3

Advanced 3GPP2 Features

Dynamic Home Agent	29
Configuring Dynamic Home Agent	29
Inter-PDSN Handoff	31
Inter-PDSN Handoff Process	31
Inter-PDSN Handoff Assignment of Home Agent and Home Address	32
Configuring Inter-PDSN Handoff	32
New Session Hotlining.....	33
Configuring New Session Hotlining.....	33
Enabling New Session Hotlining	34
Configuring the nshl.att File	34
Hotlining Prepaid Sessions.....	35
Disabling New Session Hotlining	36
Creating Hotline Profiles	36
Hotline Capability Attribute	37
Filtering Hotlining Capabilities	37
Assigning Hotline Profiles to Users.....	38
Assigning Hotline Profiles with the SQL Database	38
Populating the SQL Database with Profile Names	39
Configuring the radsqldb.aut or radsqldb.aut File	39
Assigning Hotline Profiles with Steel-Belted Radius	40
Prepaid Data Services	41
Overview	41
Components of the Prepaid Module	42
Configuring Prepaid Functionality	43
Configuring the 3gpp2.ini File for the Prepaid Module	43
Configuring Filters for Prepaid Attributes.....	44
Configuring the prepaidAcct.acc Plugin File.....	45
Configuring the prepaidAttr.att Plugin File.....	45
[Bootstrap] Section of prepaidAttr.att	46
[Settings] Section of prepaidAttr.att	46
Configuring the parlayPPSplugin.gen File	47
Determining the Volume or Duration Threshold.....	48
Configuring the radius.ini File for the Prepaid Module.....	50
Sending and Receiving Prepaid Attributes.....	50

Prepaid Attributes.....	51
Filtering Prepaid Attributes.....	53
Configuring Prepaid Timeouts.....	54
Session Timeouts.....	54
Request Timeouts.....	55
Using New Session Hotlining and Prepaid Sessions Together.....	55
Disabling Prepaid Functionality.....	56
Dynamic Mobile Update.....	56
Summary of DMU Data Exchange.....	56
Configuring DMU for Steel-Belted Radius.....	57
Configuring the 3GPP2.ini File for DMU.....	57
Configuring the dmuaut File.....	57
[Bootstrap] Section of dmuaut.....	58
[Settings] Section of dmuaut.....	58
[DatabaseQueries] Section of dmuaut.....	60
Public Key / Private Key Pair.....	61
[PKID_PrivateKey] Section of dmuaut.....	61
Configuring the sqlaccessor.gen or sqlaccessorjdbc.gen File for DMU.....	63
[Settings] Section.....	63
[VariableTypes] Section.....	64
[Query] Section.....	65
[Query/DMUSelect] Section.....	65
[Results/DMUSelect] Section.....	66
[Query/DMUUpdate] Section.....	68
Configuring the radius.ini File for DMU.....	68
Creating and Executing a Stored Procedure for DMU.....	69
Corresponding Request Types and Key Types.....	69
Stored Procedure Required Code.....	69
Stored Procedure Execution.....	70
Initializing the Database for DMU Processing.....	71
Configuring the Steel-Belted Radius for DMU.....	72
LDAP Configuration Interface Extensions.....	72
Support for HRPD Access Networks.....	72
Configuring for BCD Encoding.....	73

Chapter 4

3GPP

Overview.....	75
Product Registration.....	76
MIM Module Configuration.....	76
[Settings] Section.....	76
[Attributes] Section.....	77
LDAP Configuration Interface Enhancements.....	77

Index

About This Guide

The *Mobile IP Module Guide* describes how the Mobile IP module works and how to configure Steel-Belted Radius to support 3GPP2 or 3GPP services in the Mobile IP module. This manual is a supplement to the *Steel-Belted Radius/Service Provider Edition Administration Guide*.

Online help for Steel-Belted Radius is available by selecting the **Help** option while in the Administrator program. The topics covered in this manual do not appear in the online help for Steel-Belted Radius Administrator.

Who Should Read This Guide

This guide is intended for system and network administrators responsible for configuring mobile user authentication.

Before You Begin

Before you use this manual, you should review the *Steel-Belted Radius/Service Provider Edition Administration Guide* to gain an understanding of how the components of Steel-Belted Radius work together. You should read the Steel-Belted Radius release notes for updates about software features and requirements and corrections to your Steel-Belted Radius documentation.

This manual assumes that you have already installed and configured the Steel-Belted Radius software. For information on installing and configuring Steel-Belted Radius, refer to the, *Steel-Belted Radius Getting Started*, the *Steel-Belted Radius/Service Provider Edition Administration Guide*, and the *Steel-Belted Radius Reference Guide*.

Documentation Conventions

Convention	Indicates	Examples
Italic	Book titles, new terms, and emphasis in text.	<i>Steel-Belted Radius Administration Guide</i>
Bold	Text that guides your interaction with the software's user interface, such as commands you must enter or screen controls you must operate.	Enter Yes . Click the OK button. Choose Edit > Paste .
Bold Italic	Information you must enter in response to a screen prompt.	Enter your login name and password: Name: <i>Name</i> Password: <i>Password</i>
Courier	System messages and output, command line prompts, pathnames, filenames, and command names.	3gpp2.ini
<i>Courier Italic</i>	Variable text, where you replace a placeholder in a code sample with information appropriate to your environment.	[HAs] <i>HA-address</i> [= <i>password</i>] ...
Menu commands	Menu commands are presented as the name of the menu, followed by the > sign, and concluding with the name of the command itself. If a menu item displays a hierarchical menu, the intermediate menu selection is included in the menu path.	Choose Edit > Cut . Choose Edit > Paste As... > Text .
SMALL CAPS	Identifies a keyboard key.	Press ENTER.
Plus (+) sign	Indicates that you must press two or more keys simultaneously.	Press CTRL+B.
Square ([]) brackets	Indicates an optional parameter.	<i>HA-address</i> [= <i>password</i>]
Vertical () symbol	Indicates you should choose one keyword or variable from the list of options in command syntax.	restart { yes no }

For More Information

For information on Steel-Belted Radius, refer to *Steel-Belted Radius/Service Provider Edition Administration Guide*, *Steel-Belted Radius Getting Started Guide*, and *Steel-Belted Radius Reference Guide*

For information on 3GPP2, refer to <http://www.3gpp2.org>.

For information on 3GPP, refer to <http://www.3gpp.org>.

Chapter 1

About 3GPP2 and the Mobile IP Module

This chapter describes the concepts of 3GPP2 support in the Mobile IP Module (MIM) for Steel-Belted Radius.

The 3GPP2 feature set complies with *cdma2000 Wireless IP Network Standard, Revision C*, available at <http://www.3gpp2.org>.

NOTE: Although the Mobile IP Module for Steel-Belted Radius supports both 3GPP and 3GPP2, you cannot enable 3GPP and 3GPP2 simultaneously. If 3GPP2 and 3GPP are both enabled, Steel-Belted Radius disables 3GPP and reports the configuration error in the radius log file.

Overview

The Mobile IP Module for Steel-Belted Radius extends RADIUS functionality to 3GPP2 users on the scale required by Internet Service Providers and carriers. MIM builds on the features of Steel-Belted Radius to enable authentication of mobile users, deliver the appropriate level of service to each subscriber, and log and record all subscriber connection data for billing purposes.

3GPP2 features in the Mobile IP Module include the following:

- ▶ An implementation of the 3GPP2 Mobile IP standard, which facilitates the establishment of Mobile IP sessions by providing matching configuration information for each Mobile IP session to a *Foreign Agent (FA)* and a *Home Agent (HA)*.
- ▶ An implementation of the 3GPP2 High Rate Packet Data (HRPD) standard, which adds support for A12 interface functions to the Access-Network AAA (AN-AAA) functions supported in Steel-Belted Radius. The HRPD standard is described in the *Interoperability Specification (IOS) for High Rate Packet Data (HRPD) Access Network Interfaces* document at <http://www.3gpp2.org>.

IP Address Management

Within a wired network, Steel-Belted Radius performs authentication, authorization, and accounting of fixed data connections. With the Mobile IP Module, Steel-Belted Radius performs these tasks on mobile connections within a wireless network, where users can physically move from place to place while maintaining their connection to the network.

When a mobile node requests access to the Internet, the PDSN (Packet Data-Serving Node) sends an authentication request to the Steel-Belted Radius server. A PDSN provides access to the Internet, intranets and Wireless Application Protocol servers for mobile stations utilizing a CDMA2000 Radio Access Network (RAN). The server verifies the user's credentials, and sends an Access-Accept message along with information about how to configure the connection to the PDSN, which then routes the user onto the Internet.

In such a wireless network, the user's mobile node is provided with a durable IP address that can persist, even as point-to-point radio connections are made and broken as the user moves from cell to cell in the wireless provider's network and across multiple networks.

A wireless operator can offer the following two levels of IP address management:

- ▶ Simple IP (SIP)
- ▶ Mobile IP (MIP)

NOTE: See *“Home Address Assignment Process” on page 14* for more information about IP address management.

Simple IP

Simple IP provides a low level of IP address mobility. Simple IP requires that mobile users remain associated with the same PDSN in order for mobile nodes to retain the same IP address. As long as the user moves among the cells associated with the PDSN, the PDSN can keep track of the mobile node and assign it the same IP address each time it reconnects through a new cell. Simple IP is similar from IP addressing in a wired connection, where one address maps to one connection.

Simple IP guarantees IP address mobility only within a restricted geographic coverage area, and only within the network of a single provider. If the user moves to an area handled by a different PDSN, the previous connection is terminated, another connection is established, and a new IP address is assigned.

Subscribers might choose to use Simple IP if they have older handsets that are not loaded with software that supports Mobile IP. Additionally, Simple IP may be a lower-cost option offered by wireless providers.

Mobile IP

Mobile IP provides a higher level of mobility than Simple IP. A subscriber is able to move from place to place and maintain the same connection and IP address. If the subscriber moves outside of the geographic area of the home network, the subscriber may possibly roam to another provider's network if there is a roaming agreement.

When Mobile IP is used, the user's wireless provider maintains a Home Agent (HA) through which all the user's traffic is routed. The user's mobile node connects to the network via an PDSN, which functions as a Foreign Agent (FA).

The FA contacts the HA on behalf of the user. The HA is able to make the user's IP address appear constant to any party with which the user is in contact by simply routing the mobile node's traffic to the FA at the user's current geographical position. The Steel-Belted Radius server plays a central role in supporting a Mobile IP infrastructure by coordinating communication between HAs and FAs, and by controlling the potential security association between those devices.

IP Address Durability in Roaming Scenarios

Roaming occurs when one provider's subscriber connects through another provider's network. Roaming can occur for both Simple IP and Mobile IP.

When the mobile node requests access to the network from a roaming partner, the PDSN sends an authentication request to its local Steel-Belted Radius server, which proxies (forwards) it to the home server on the subscriber's provider network. That Steel-Belted Radius server verifies the user's credentials and sends an Access-Accept message along with information about how to configure the connection (which may include an assigned IP address and HA address), allowing the PDSN to route the user onto the network.

With Simple IP, when a mobile node crosses from one provider network to another, its old AAA session is closed, a new one is established via the new PDSN, and the user is issued a new IP address (assuming the RADIUS server is in charge of assigning IP addresses).

With Mobile IP, when the mobile node requests access to the Internet from a roaming partner, the process is almost the same as that for Simple IP. The primary difference is that there is an additional, intermediate local server that proxies requests to the partner network. The foreign AAA server (FAAA) proxies (forwards) the PDSN's request to the appropriate home AAA server (HAAA) at the user's home provider network. The FAAA is able to determine the correct HAAA to be used in the connection based on user name decoration or other available information.

The Mobile IP session that requires roaming is set up in the same way as in the non-roaming case, except that, in this case, a tunnel is provided between the PDSN on the partner network and the HA on the provider network. The HAAA is responsible for any security association between the FA and HA. The tunnel hides the fact that the user is mobile and allows the user to access the network through the appropriate HA.

NOTE: A Steel-Belted Radius server without MIM can act as a FAAA server. HAAA functionality requires a Steel-Belted Radius server and the MIM feature.

Support for Special Accounting Requests

The 3GPP2 specification allows for a PDSN to generate multiple Accounting Start/Stop pairs for a given session (this is true for SIP and MIP sessions). When a PDSN sends an Accounting Stop request and includes a 3GPP2-Session-Continue attribute with a value of 1 in the request, the PDSN is indicating to the AAA server that the session should not be deleted. When it receives such a request, the Steel-Belted Radius server with MIM

marks the session as dormant and does not free any of its allocated resources (such as the IP address).

This type of Accounting Stop request is typically followed by an Accounting Start that has the effect of marking the session active again.

When a PDSN sends an Accounting Stop request and includes a 3GPP2-Session-Continue attribute with a value of 0 in the request, the PDSN is indicating to the AAA server that the session should be deleted. A new session may be created for a different PDSN. For example, this transfer to a different PDSN typically occurs in the case of inter-PDSN handoff as explained in [“Inter-PDSN Handoff”](#) on [page 31](#).

Chapter 2

3GPP2 Mobile IP Module Configuration

This chapter explains the determination of request types by Steel-Belted Radius, how Home Agent and Home Address are assigned, and configuration requirements for Mobile IP using 3GPP2.

Product Registration

The Mobile IP Module is fully integrated into Steel-Belted Radius. If you have purchased MIM, a single license key activates both Steel-Belted Radius and MIM. When you install Steel-Belted Radius, the system requests that you type in the license key. After the license key is registered, you should edit the configuration settings as explained in the sections in this chapter.

Simple IP Configuration

This section provides background information about Simple IP (SIP) configuration.

Simple IP Request Types

The AAA server must be able to handle two types of requests: SIP User Authentication and Other. A SIP User Authentication request is sent by the PDSN when it needs to authenticate a user. SIP support within MIM is limited to specifying a filter to be applied to SIP User Authentication requests and handling the 3GPP2-Session-Continue attribute on resulting accounting requests.

The Steel-Belted Radius server first determines whether an Access-Request is a SIP User Authentication or Other request in the following manner (also see [Figure 1 on page 10](#)):

- 1 If the request contains a 3GPP2-Correlation-Id attribute, the request is a SIP User Authentication request.
- 2 Otherwise it is an Other request.

NOTE: Requests for SIP and MIP may be handled by the same AAA server. When handling for both types of requests is enabled, all MIP-related request classification checks are performed prior to the check for SIP User Authentication requests.

SIP User Authentication Requests

In a SIP scenario, the PDSN sends a request to authenticate the end user to the AAA server.

Steel-Belted Radius classifies Access-Requests as SIP as shown in [Figure 1 on page 10](#).

If Steel-Belted Radius is not configured to accept this type of request, it rejects the request.

If the feature is enabled, the request is processed as follows:

- 1 The user is authenticated normally, and, if successful, Steel-Belted Radius generates attributes for an Access-Accept response (from the return list, authentication plug-in, etc.).
- 2 A configured filter is applied to the attributes in the Access-Accept response, and the response is sent.

Mobile IP Configuration

This section explains how Mobile IP request types are determined. Identification of the request type allows you to know what attributes the Home AAA server needs to receive from the PDSN.

Mobile IP Request Types

The PDSN or HA may send five different types of requests to the Home AAA server. Access-Requests must be configured correctly, based on the type of request. The configuration for each type of request is described in the [“Mobile IP Module Configuration” on page 20](#).

The types of requests are:

- ▶ Authorize Only
- ▶ FA User Authentication
- ▶ HA Key Distribution
- ▶ MN-HA Shared Key Distribution
- ▶ HA User Authentication
- ▶ Other

This section describes each type of request to help you identify the request. This section also describes the configuration requirements for the Home AAA server that enable it to handle the incoming request of each type.

- ▶ **Authorize Only request**

An Authorize Only request (also referred to as an Online Prepaid request) is used to

obtain additional quota from the prepaid server. For more information about the Prepaid module, see “Prepaid Data Services” on page 41.

▶ **FA User Authentication request**

An FA User Authentication request is sent by the FA to authenticate a user and get Mobile IP configuration and authorization information. The information the FA requires may include the address of the appropriate HA and a Pre-Shared-Secret for use in keying an IPsec (IP security) tunnel between the FA and HA. Based on this information, the FA communicates with the HA and the FA notifies the HA of the presence of the user on the foreign (FA) network.

▶ **HA Key Distribution request**

An HA Key Distribution request is sent by the HA whenever it needs to retrieve the *S-Key (Shared-Key)* that is used to construct Pre-Shared-Secrets for an IPsec tunnel being initiated by a particular FA.

The HA does not send an HA Key Distribution request if the FA did not request a Pre-Shared Secret in its request to the Steel-Belted Radius server. In this case, no IPsec is used to encrypt communications between the FA and HA and the HA does not need any keys.

Even when IPsec is used to protect the communication from FA to HA, the HA does not need to retrieve an S-Key each time a mobile station sends a mobile registration request. It needs to send an HA Key Distribution request only when it does not have a fresh S-Key to use in communicating with a particular FA. The S-Lifetime attribute in the Access-Accept response indicates how soon the S-Key needs to be refreshed.

▶ **MN-HA Shared Key Distribution request**

An MN-HA Shared Key Distribution request is sent by the HA when it needs to authenticate a user whose mobile node does not support some of the newer authentication extensions in its mobile registration request. When Steel-Belted Radius processes this request, it retrieves the user's password from a database or directory and sends it in encrypted form to the HA. The HA, in turn, decrypts the password and authenticates the user.

▶ **HA User Authentication request**

An HA User Authentication request is sent by the HA when it needs to authenticate a user whose mobile node does support the new authentication extensions in its mobile registration request. When Steel-Belted Radius processes this request, it checks the user's credentials and accepts or rejects the user based on the outcome.

▶ **Other request**

A request of type Other is one that does not fall into any of the above categories. This would typically be a request unrelated to Simple IP or Mobile IP in a 3GPP2-compliant network.

Determining Mobile IP Request Types

The home RADIUS server first determines the Access-Request type (Authorize Only, FA User Authentication, HA Key Distribution, MN-HA Shared Key Distribution, HA

User Authentication, or Other). The process for determining the type of Access-Request is illustrated in [Figure 1 on page 10](#) and described below:

The request is an Authorize Only request if:

The request contains Service-Indicator attribute with a value of Authorize-Only.

The request is an FA User Authentication request if:

- a The request contains a 3GPP2-Pre-Shared-Secret-Request attribute with a value of 1
- or
- b a 3GPP2-Home-Agent-Address attribute (of any value).

The request is an MN-HA Shared Key Distribution request if:

- a The request contains a 3GPP2-MN-HA-SPI attribute
- and
- b The MN-HA Shared Key Distribution feature is enabled.

The request is an HA Key Distribution request if:

- a The Use-S-Request-As-Marker setting is set to 1 in the `3gpp2.ini` [HA-Key-Distribution-Requests] section.
- and
- b The request contains a 3GPP2-S-Request attribute with a value of 1.
- and
- c The User-Name consists of (a) a concatenation of two IP addresses in dotted notation, separated by a period, and the second address corresponds to a configured HA address; or (b) the User-Name consists of an ASCII hexadecimal notation in which both the FA address and HA address are converted to hexadecimal. (See [“HA Key Distribution Requests” on page 17](#) for more information.)

NOTE: *If the Use-S-Request-As-Marker setting is set to 0, then the request does not need to contain the 3GPP2-S-Request attribute, but must still meet the requirement for the User-Name described in (c) above.*

The request is an HA Key Distribution request if:

- a The feature that enables scanning for the 3GPP2-S-Request attribute is not enabled
- and
- b The User-Name is either (a) a concatenation of two IP addresses in dotted notation, separated by a period, and the second address corresponds to a configured HA address; or (b) an ASCII hexadecimal notation in which both the FA address and HA address are converted to hexadecimal. (See [“HA Key Distribution Requests” on page 17](#) for more information.)

The request is an HA User Authentication request if:

- a The request contains a 3GPP2-Correlation-Id attribute
and
- b The request originated from an HA (a list of HAs is provided in the `3gpp2.ini` file and Either the source IP address or NAS-IP-Address attribute can be used to check for a match).

The request is a SIP User Authentication request if:

- a The request contains a 3GPP2-Correlation-Id attribute
and
- b The request did not originate from an HA.

Otherwise, it is an Other request.

For HA Key Distribution requests, additional checks may be performed (as configured) to ensure that the HA address corresponds to the source address of the request and/or the NAS-IP-Address in the request. The HA may even be required to authenticate itself against a fixed password or one configured into the Steel-Belted Radius Native User database.

[Figure 1 on page 10](#) illustrates the decision process for determining whether an Access-Request is an FA User Authentication, HA Key Distribution, MN-HA Shared Key Distribution, HA User Authentication, or Other request.



Figure 1 Access Type Determination

FA User Authentication Requests

NOTE: `Accept-Requests=1` *must appear in the [FA-User-Auth-Requests] section of the 3gpp2.ini file to enable FA.*

An FA User Authentication request is expected to contain the following attributes:

- ▶ The User-Name attribute.
- ▶ The CHAP-Challenge and CHAP-Password attributes.
- ▶ The NAS-IP-Address attribute containing the address of the FA.
- ▶ The 3GPP2-Home-Agent-Address attribute containing the address of the desired HA, or 0.0.0.0 or 255.255.255.255 to request that an HA be assigned.
- ▶ The 3GPP2-Correlation-Id attribute containing a session string that is present in authentication as well as accounting requests. 3GPP2-Correlation-Id identifies a user session that may span multiple RADIUS accounting start/stop pairs.
- ▶ The optional 3GPP2-Pre-Shared-Secret-Request attribute containing the value 1 to request a 3GPP2-Pre-Shared-Secret, or 2 (or not present) to not request one.

An Access-Accept response is returned to the FA (on successful end-user authentication). Steel-Belted Radius generates attributes for an Access-Accept (from the return list, authentication plug-in, etc.). In addition to attributes configured for the user, the response includes the following attributes:

- ▶ A 3GPP2-Home-Agent-Address attribute indicating the address of the HA with which to set up a session.
- ▶ A 3GPP2-Pre-Shared-Secret attribute, if requested, containing the secret shared with the HA for tunnel establishment. This attribute is salt-encrypted.
- ▶ If 3GPP2-Pre-Shared-Secret-Request is 1 (a Pre-Shared-Secret is requested,) the S-Key is looked up based on the HA address.
 - ▷ If no S-Key is found, or if the S-Key's lifetime has expired, a new S-Key is generated for this HA.
 - ▷ The 3GPP2-Pre-Shared-Secret and 3GPP2-Key-ID attributes are added to the Access-Accept response
- ▶ The 3GPP2-Pre-Shared-Secret and 3GPP2-Key-ID attributes are added to the Access-Accept response.
- ▶ If 3GPP2-Pre-Shared-Secret-Request is 2 or if the attribute is not present in the request, the response packet does not contain security information attributes (3GPP2-Pre-Shared-Secret and 3GPP2-Key-ID).
- ▶ The configured filter is applied to the attributes in the Access-Accept response, and the response is sent.

NOTE: *The Home RADIUS server maintains the current S-Key and its lifetime for each HA. Whenever an FA User Authentication or HA Key Distribution request is received and the S-Key for the HA has not yet been generated or has expired (according to its associated lifetime value), a new S-Key is generated for the HA.*

Home Agent (HA) Assignment Process

The assigned Home Agent address is contained in the 3GPP2-Home-Agent-Address attribute of the FA User Authentication Access-Accept. The assignment of the Home Agent address depends on three basic factors:

- ▶ The value of the 3GPP2-Home-Agent-Address attribute in the Access-Request.
- ▶ The user's configured HA-Address in the return list within Steel-Belted Radius.
- ▶ Whether or not Dynamic HA assignment (assignment of a Home Agent from a weighted pool of addresses) is enabled or disabled.

The Home Agent address assignment process is described in the following list. [Table 1 on page 13](#) summarizes HA address assignment.

- 1 If the 3GPP2-Home-Agent-Address attribute is not present in the return list (or if it is present and its value is 0.0.0.0), the value for the response 3GPP2-Home-Agent-Address attribute is set to the value of the 3GPP2-Home-Agent-Address attribute contained in the request.
- 2 If the value for 3GPP2-Home-Agent-Address in the configured return list is 255.255.255.255, the value of the response 3GPP2-Home-Agent-Address attribute is set to the value of the 3GPP2-Home-Agent-Address attribute contained in the request, as long as the received value is a configured and recognized HA address; otherwise, the user is rejected. (The list of acceptable HA addresses is configured in the [HAs] section of `3gpp2.ini`.)
- 3 If a specific value for 3GPP2-Home-Agent-Address in the Access-Request is the same as the value in the configured return list, the requested address is assigned.
- 4 If the value for 3GPP2-Home-Agent-Address in the Access-Request is a specific address and it is not equal to the value of 3GPP2-Home-Agent-Address in the return list, then, depending on configuration of HA-Address-Mismatch in the `3gpp2.ini` file, one of the following occurs:
 - ▷ The request is rejected.
 - ▷ The response 3GPP2-Home-Agent-Address is set to the request 3GPP2-Home-Agent-Address.
 - ▷ The value of 3GPP2-Home-Agent-Address in the return list is returned.
- 5 If the value for 3GPP2-Home-Agent-Address in the Access-Request is 0.0.0.0 or 255.255.255.255, Steel-Belted Radius assigns a Home Agent based on the following:
 - ▷ If Dynamic HA assignment is enabled, the response HA address is taken from the configured weighted round robin pool of HA addresses. "Enabled" means that the line containing HA-Address-Round-Robin-Group is present in the `3gpp2.ini` file. See "[Configuring Dynamic Home Agent](#)" on page 29.
 - ▷ If Dynamic HA assignment is not enabled, the response HA address is taken from the configured return list.
- 6 If a request for inter-PDSN handoff is detected and inter-PDSN handoff is enabled, the session maintains the same Home Agent. See "[Inter-PDSN Handoff](#)" on page 31.

Home Agent (HA) Assignment Summary

Table 1 summarizes the factors that affect Home Agent assignment. (The HA Address in the Access-Request is the value of the 3GPP2-Home-Agent-Address attribute.)

Table 1. Home Agent Assignment Factors

HA Address in Access-Request	Dynamic HA Assignment	Configured HA Address in the Return List	HA Assignment
0.0.0.0 or 255.255.255.255	Enabled	N/A	HA assigned from the round robin group. (See “Dynamic Home Agent” on page 29 for more information about Round-Robin Load Balancing.)
0.0.0.0 or 255.255.255.255	Disabled	Specific HA address	Taken from the configured return list.
0.0.0.0 or 255.255.255.255	Disabled	0.0.0.0 or 255.255.255.255 or none	Access accepted and no HA assigned.
Specific HA address	N/A	0.0.0.0	HA assigned as requested without any checking in the list of acceptable HA addresses. (Acceptable HA addresses are configured in the [HAs] section of <code>3gpp2.ini</code> , as described in the “[HAs] Section” on page 27 .)
Specific HA address	N/A	255.255.255.255	Requested HA is assigned if it is one of the list of acceptable HA addresses. (Acceptable HA addresses are configured in the [HAs] section of <code>3gpp2.ini</code> , as described in the “[HAs] Section” on page 27 .) Otherwise, the Access-Request is rejected.
Specific HA address	N/A	Specific HA address	If the requested HA matches the configured HA, the requested HA is assigned. If there is a mismatch, the assignment complies with the setting of the HA-Address-Mismatch rule. (See “[FA-User-Auth-Requests] Section” on page 23 .)
Inter-PDSN detected and enabled: Access-Request contains an HA address of 0.0.0.0 or 255.255.255.255 or the specific HA already in use for the session*	N/A	Any	The existing HA address is assigned so that the session maintains the same HA as originally assigned.
Inter-PDSN handoff detected and enabled: Access-Request contains a specific HA address that differs from the HA already in use for the session*	N/A	Any	Access-Request is rejected.

* If inter-PDSN handoff is not enabled, the Access-Request is treated as a new access request. See “[Inter-PDSN Handoff](#)” on [page 31](#) for more information.

Home Address Assignment Process

The assigned Home Address is returned in the Framed-IP-Address attribute of the FA User Authentication Access-Accept. Steel-Belted Radius assigns a Home Address to a mobile node requesting network access.

The assignment of the Home Address depends on two factors:

- ▶ The value of the Framed-IP-Address attribute in the Access-Request. (The value can be 0.0.0.0, or 255.255.255.255, or a specific IP address.)
- ▶ The value of the FramedIPAddressHint field in the `radius.ini` file. (You can set this value to Yes, No, or Check-Pool.)

The Home Address assignment process is described in the following list. [Table 2 on page 15](#) summarizes Home Address assignment.

- 1 If the value of Framed-IP-Address in the Access-Request is a specific IP address (not 0.0.0.0 or 255.255.255.255), Steel-Belted Radius then assigns the Home Address based on the configured value of FramedIPAddressHint.
 - ▶ If the value of FramedIPAddressHint is Yes, the requested Home Address is assigned. (Value of Yes is not recommended; use Check-Pool.)
 - ▶ If the value of FramedIPAddressHint is No, the Home Address is taken from the configured pool of addresses for Framed-IP-Address. The next available address is used. (Value of No is not recommended; use Check-Pool.)
 - ▶ If the value of FramedIPAddressHint is Check-Pool (recommended), the requested Home Address is checked for validity against the pool of addresses for Framed-IP-Address.
 - ▷ If the requested address is within the pool (valid) and not already in use, the requested Home Address is assigned.
 - ▷ If the requested address is not within the pool (not valid), the Access-Request is rejected.
 - ▷ If the requested address is within the pool but is already in use, the requested address is assigned and the internal reference count for that address is increased by one to keep track of the number of times the address is used. When all sessions using that address are released, the address is returned to the pool.
- 2 If the value of Framed-IP-Address in the Access-Request is 0.0.0.0 or 255.255.255.255, the Home Address is taken from the configured pool of addresses for Framed-IP-Address. The next available address is used. It does not matter if FramedIPAddressHint is Yes, No, or Check-Pool.
- 3 If a request for inter-PDSN handoff is detected (an existing session requests access), the session maintains the same Home Address if the Home Address in the Access-Request is 0.0.0.0 or 255.255.255.255. If the Home Address in the Access-Request is a specific address and it does not match the address in the

configured response, the request is rejected. See “[Inter-PDSN Handoff](#)” on page 31 for more information.

Home Address Assignment Summary

Table 2 summarizes the factors that affect Home Address assignment. (The Home Address in the Access-Request is the value of the Framed-IP-Address attribute.)

Table 2. Home Address Assignment Factors

Home Address in Access-Request	FramedIPAddress Hint Setting in radius.ini	Home Address Assignment
Specific Home Address	Yes (not recommended)	Same Home Address assigned as requested.
Specific Home Address	No (not recommended)	Home Address is assigned from the pool of Home Addresses.
Specific Home Address	Check-Pool	<p>Requested Home Address is checked for validity against the pool of addresses for Framed-IP-Address.</p> <p>If the requested address is valid, it is assigned.</p> <p>If the requested address is valid but in use, it is assigned and the internal reference count for that IP address is increased by one. The address is returned to the pool when it is released by all sessions.</p> <p>If the requested address is not valid, the Access-Request is rejected.</p> <p>Example:</p> <p>radius.ini file</p> <pre>[Configuration] FramedIpAddressHint=check-pool</pre> <p>For information about setting up IP address pools, see the <i>Steel-Belted Radius Administration Guide</i>.</p>
0.0.0.0 or 255.255.255.255	Yes, No, or Check-Pool	Home Address is assigned from the pool of Home Addresses.
Inter-PDSN detected and enabled: Home-Address in the Access-Request is 0.0.0.0 or 255.255.255.255. *	Check-Pool (Yes or No are not valid for Inter-PDSN handoff.)	<p>The original Home Address is assigned so that the session maintains the same Home Address as originally assigned. (See “Inter-PDSN Handoff” on page 31 for more information.)</p> <p>NOTE: The request is rejected if the requested address is a specific address that does not match the configured Home Address in the return list.</p>
Inter-PDSN detected and enabled: Access-Request contains a specific Home Address that differs from the HA already in use for the session*	N/A	Access-Request is rejected.

*If Inter-PDSN handoff is not enabled, the Access-Request is treated as a new access request. See “[Inter-PDSN Handoff](#)” on page 31 for more information.

NOTE: *If there is a specific home address in the configured return list, the configured address overrides the FramedIPAddressHint and the configured address is assigned, regardless of the requested address.*

IPsec Security Attributes Needed for FA/HA Communications

The HAAA server is responsible for any IPsec association between the FA and HA needed for tunnel establishment. The tunnel hides the fact that the user is mobile and allows the user to access the network through the appropriate HA.

Prior to RADIUS communications with the HAAA server, an FA does not usually know if the HA with which it needs to communicate requires or even supports IPsec security. Thus, the 3GPP2-Pre-Shared-Secret-Request attribute, sent from the PDSN to the HAAA in the FA User Authentication request, should have a value of 1 to indicate that security information is requested.

NOTE: *The FA may be aware of the requirement for IPsec if the user's mobile node knows the 3GPP2-Home-Agent-Address before the call flow even reaches the FA. When the FA is pre-informed of the HA's IP address, it can make a more informed response (based on information available to the PDSN) regarding the use of IPsec prior to communications with the Home RADIUS server.*

The RADIUS Access-Accept message automatically includes the 3GPP2-Pre-Shared-Secret-Request attribute and the 3GPP2-Key-ID attribute. When IPsec is needed for tunnel establishment between the FA and the HA, some PDSN (FA) vendors may require that the RADIUS Access-Accept packet contain values for all three security-related attributes: 3GPP2-Pre-Shared-Secret, 3GPP2-Key-ID, and 3GPP2-Security-Level. If one of these three values is not present in the response packet, the call may be dropped by the PDSN.

Refer to your PDSN documentation regarding this requirement for inclusion of all three security-related attributes. If the 3GPP2-Security-Level attribute is required for your PDSN, you should add it via the filter specified in the Filter setting in the [FA-User-Auth-Requests] section of the `3gpp2.ini` file.

See “[FA-User-Auth-Requests] Section” on page 23 for information about adding the filter to the configuration file. See the *Steel-Belted Radius Administration Guide* for more information about creating filters to add attributes.

Proxy Realm Based Configuration

Some networks may require a RADIUS proxy to authenticate a user's credentials for an FA User Authentication request while also requiring the HAAA server (with the installed MIM module) to manage the S-Keys needed for FA-to-HA communications. Steel-Belted Radius accommodates this scenario by not requiring the authentication of the end-user to be performed by a local authentication method.

Furthermore, if Steel-Belted Radius/Service Provider Edition is configured to use proxy realms, FA User Authentication request processing options can be configured on a realm-by-realm basis. See the *Steel-Belted Radius Administration Guide* for more information about configuring proxy realms.

HA Key Distribution Requests

An HA Key Distribution Access-Request is expected to have the following attributes:

- ▶ The User-Name attribute, containing either of the following:
 - ▷ an “ASCII “dotted-quad” representation of the FA address concatenated with a “dotted-quad” representation of the HA address, separated by a period. For example, 10.10.10.11.92.64.10.1.
 - or
 - ▷ an ASCII hexadecimal notation in which both the FA address and HA address are converted to hexadecimal. For example, an FA address of 10.10.10.11 is concatenated with an HA Address of 92.64.10.1 to yield 0A0A0A0B5C400A01.
- ▶ The User-Password or CHAP-Password value associated with the HA.
- ▶ The 3GPP2-S-Request attribute with a value of 1 (required only if Use-S-Request-As-Marker in the [HA-Key-Distribution-Requests] section of `3gpp2.ini` is set to 1).
- ▶ The NAS-IP-Address attribute (required only if Check-NAS-IP-Address in the [HA-Key-Distribution-Requests] section of `3gpp2.ini` is set to 1).

It is the HA itself, not a mobile user, that may be authenticated by Steel-Belted Radius when this type of request is received from the HA. If the HA is to be authenticated (optional, rather than required), the password information in the Access-Request must reflect the password assigned to the HA.

NOTE: For more information, see “[HAs] Section” on page 27.

An HA Access-Request is processed as follows:

- 1 The configured HA is looked up based on HA address parsed from User-Name.
- 2 If the HA address parsed from the User-Name cannot be matched to an entry in the RAS Clients portion of the internal database, the request is rejected.
- 3 If the option to check the source IP address of the Access-Request against the claimed HA address is enabled, such a check is performed. If the check fails, the request is not considered a valid HA Key Distribution request.
- 4 If the option to check the NAS-IP-Address inside the Access-Request against the claimed HA address is enabled, the NAS-IP-Address is compared to the claimed HA address. If no NAS-IP-Address is present in the Access-Request or the value of the attribute does not match the HA address, the request is not considered a valid HA Key Distribution request.
- 5 If password checking is configured for the HA, the request is authenticated and rejected if the password is incorrect.
- 6 If no password is configured for the HA, then no verification of credentials beyond the standard RADIUS packet processing and the optional HA checking in items 3 and 4 is performed.
- 7 The S-Key for this HA is retrieved. If no S-Key key is found, or if the S-Key has expired, a new S-Key is generated for this FA.

- 8 An Access-Accept is issued, which includes the 3GPP2-S-Key and 3GPP2-S-Lifetime attributes. The S-Lifetime is the time at which the S-Key expires. The S-Key is **not** encrypted.
- 9 A configured filter is applied to the attributes in the Access-Accept response, and the response is sent.

NOTE: The Home RADIUS server maintains the current S-Key and its lifetime for each HA. Whenever an FA or HA request is received and the S-Key for the HA has not yet been generated or has expired (according to its associated lifetime value), a new S-Key is generated for the HA.

MN-HA Shared Key Distribution Requests

After the FA-to-HA communication channel is in place, the FA forwards the user's mobile registration request to the HA. The HA may authenticate this mobile registration request in one of three ways:

- ▶ By sending an Access-Request to authenticate information in the mobile registration request. This request is treated as an HA User Authentication request, the handling of which is described in the next section.
- ▶ By sending an Access-Request that retrieves the MN-HA Shared Key. The request is treated as an MN-HA Shared Key Distribution request where the User-Name identifies the user for whom a password is to be retrieved. After it has received the password, the HA authenticates the user.
- ▶ By authenticating the user directly against a database or directory of users that also is used by the Steel-Belted Radius server to process FA User Authentication requests. In this case, the Steel-Belted Radius server does not receive another Access-Request from the HA.

An MN-HA Shared Key Distribution Access-Request is expected to have the following attributes:

- ▶ The User-Name attribute, identifying the end user for whom a password is to be retrieved.
- ▶ A User-Password or CHAP-Password value whose contents are ignored.
- ▶ The MN-HA-SPI attribute that identifies this as an MN-HA Shared Key Distribution request. The value of this attribute is ignored.

If Steel-Belted Radius is not configured to accept this type of request, it treats the request as being of type Other, which most likely results in it being rejected.

If processing of MN-HA Shared Key requests is enabled, the request is processed as follows:

- 1 The request to look up the password for the user identified by the User-Name attribute is passed to the configured authentication methods.
- 2 If none of the configured authentication methods can retrieve a clear-text or reversibly encrypted password for the user, the request is rejected.
- 3 If the retrieved password is reversibly encrypted, it is decrypted into its clear-text form.

- 4 An Access-Accept that includes the 3GPP2-MN-HA-Shared-Key attribute is issued. The Shared-Key is salt-encrypted if the attribute definition in the `radius.dct` specifies salt-encryption.
- 5 A configured filter is applied to the attributes in the Access-Accept response, and the response is sent.

Warning: *This feature allows user passwords to be retrieved by any configured RADIUS client. Depending on your network architecture and the trust you place in administrators of other RADIUS servers from which your server receives requests, this may represent a major security exposure.*

HA User Authentication Requests

Upon receipt of the user's mobile registration request, the HA may send an HA User Authentication request to the HAAA.

NOTE: `Accept-Requests=1` *must appear in the [HA-User-Auth-Requests] section of the `3gpp2.ini` file to enable HA.*

An HA User Authentication Access-Request is expected to have the following attributes:

- ▶ The User-Name attribute, which identifies the end user for whom a password is to be retrieved.
- ▶ A User-Password or CHAP-Password value, which is used to authenticate the user.
- ▶ The 3GPP2-Correlation-Id attribute, which identifies this as an HA User Authentication request.

If Steel-Belted Radius is not configured to accept this type of request, it treats the request as having type Other, resulting in a different filter being applied to the final response.

If the feature is enabled, the request is processed as follows:

- 1 The end-user is authenticated normally, and, if successful, Steel-Belted Radius generates attributes for an Access-Accept (from the return list, authentication plug-in, etc.).
- 2 A configured filter is applied to the attributes in the Access-Accept response, and the response is sent.

IP Address Assignments from Filters

Using Steel-Belted Radius and MIM, attribute filters can be used to assign IP addresses from IP address pools managed by the Steel-Belted Radius server. The filters typically reference a specific fixed pool name, though problems may occur if a single HAAA is responsible for multiple HAs and each HA is expecting end-user IP addresses to be assigned from a different range.

Instead of a fixed pool name, a special pool name of `<pool associated with RAS Client>` can be specified in the filter. This causes an IP address to be assigned from the pool associated with the RADIUS client entry in the database from which the request was

received. You can then add a RADIUS client entry for each HA and assign each one a unique IP address pool.

Attributes and Filters

Each mobile user must be configured with return list attributes to satisfy the various types of requests (FA User Authentication, MN-HA Key Distribution, HA User Authentication or Other) that the Steel-Belted Radius server may need to process. Since each type of request may require a different subset of these attributes, optional filters may be specified to modify the actual attributes returned in response to each type of request.

HA Key Distribution requests are the only type of request that do not involve a specific mobile user and are handled entirely by code with no lookup of return list attributes. It is nevertheless possible to configure an optional filter to add additional attributes that your HA may require.

Mobile IP Module Configuration

This section summarizes the sections and settings of the `3gpp2.ini` file that control the functionality of the 3GPP2 feature set.

NOTE: *The `3gpp2.ini` file is read whenever Steel-Belted Radius restarts or receives a HUP signal.*

The `3gpp2.ini` file, which contains the settings for the Mobile IP Module, consists of the sections described in [Table 3](#).

Table 3. *3gpp2.ini* Sections

Section	Description
[Settings]	Enables or disables MIM processing.
[Authorize-Only-Requests]	Enables Prepaid online requests. See “[Authorize-Only-Requests] Section” on page 22 for more information.
[FA-User-Auth-Requests]	Specifies options for processing FA User Authentication requests. See “[FA-User-Auth-Requests] Section” on page 23 for more information.
[HA-Key-Distribution-Requests]	Specifies options for processing HA Key Distribution requests. See “[HA-Key-Distribution-Requests] Section” on page 23 for more information.
[MN-HA-Shared-Key-Requests]	Specifies options for processing MN-HA Shared Key requests. See “[MN-HA-Shared-Key-Requests] Section” on page 25 for more information.
[HA-User-Auth-Requests]	Specifies options for processing HA User Authentication requests. See “[HA-User-Auth-Requests] Section” on page 25 for more information.

Table 3. *3gpp2.ini Sections (Continued)*

[SIP-User-Auth-Requests]	Specifies options for processing SIP User Authentication requests. See “[SIP-User-Auth-Requests] Section” on page 26 for more information.
[Other-Requests]	Specifies options for processing other (none of the above) requests. See “[Other-Requests] Section” on page 26 for more information.
[Attributes]	Identifies dictionary attributes necessary for request processing. See “[Attributes] Section” on page 27 for more information.
[HAs]	Lists the identities of HAs and whether and how they should be authenticated. See “[HAs] Section” on page 27 for more information.

The `3gpp2.ini` file may also contain one or more sections named `[FA-User-Auth-Requests/name]`, where *name* is the value of a `FA-User-Auth-Requests-Section` setting in the `[3gpp2]` section of a proxy realm (`.pro`) configuration file.

The `radius.dct` file shipped with Steel-Belted Radius has been configured with the attributes necessary for supporting mobile IP services in compliance with the 3GPP2 standards.

[Settings] Section

The `[Settings]` section (Table 4) contains the master switch that enables the MIM 3GPP2 feature set.

Table 4. *[Settings] Section*

Setting	Description
Enable	Set this field to 1 to enable support for 3GPP2 features. Specifying 0 disables this module.
S-Seconds	Set to the lifetime, in seconds, for each new S-Key that the server generates. After this many seconds, a new S-Key is generated. The S-Key is used in processing FA User Authentication and HA Key Distribution requests. This value also defines the frequency with which an HA must make new HA Key Distribution requests for communicating with FAs.
AddFunk3GPP2RequestTypeToRequest	This setting is related to the Dynamic Mobile Update feature. Set to 1 to add the Funk-3GPP2-Request-Type attribute to the request, enabling the appropriate key password to be obtained from the database. (See “Dynamic Mobile Update” on page 56 and “Corresponding Request Types and Key Types” on page 69 for more information.)

Table 4. [Settings] Section (Continued)

Setting	Description
MIPSessionsForDevice	Inter-PDSN Handoff field. See “Inter-PDSN Handoff” on page 31 for more information.
MustHaveSameHomeAndHAAddresses	Set to 1 (default) to enable Inter-PDSN handoff. Comment out or set to 0 to disable Inter-PDSN handoff.
UniqueDeviceIdentifier	Inter-PDSN Handoff field for identifying how a session is determined to be an existing session. If the User-Name, Calling-Station-Id, or both for an Access-Request match an existing session, the session is considered to be an existing session. See “Inter-PDSN Handoff” on page 31 for more information. Valid values are: <ul style="list-style-type: none"> • User-Name • Calling-Station-Id • User-Name, Calling-Station-Id

Example

```
[Settings]
Enable = 1
S-Seconds = 3600
AddFunk3GPP2RequestTypeToRequest = 1
MIPSessionsForDeviceMustHaveSameHomeAndHAAddresses=1
UniqueDeviceIdentifier=User-Name, Calling-Station-Id
```

[Authorize-Only-Requests] Section

The [Authorize-Only-Requests] section ([Table 5](#)) specifies how Prepaid online requests should be handled. For more information about the Prepaid module, see [“Prepaid Data Services” on page 41](#).

Table 5. [Authorize-Only-Requests] Section

Setting	Description
Accept-Requests	Must be set to 1 to allow the Prepaid module to function.
Filter	The name of the filter applied to attributes in an Access-Accept that is issued in response to an FA User Authentication request. The filter must be defined in the <code>filter.ini</code> file. If no filter is specified, all attributes are returned unchanged.

[FA-User-Auth-Requests] Section

The [FA-User-Auth-Requests] section (Table 6) specifies how FA User Authentication requests should be handled.

Table 6. [FA-User-Auth-Requests] Section

Setting	Description
Accept-Requests	<ul style="list-style-type: none"> If set to 0, FA User Authentication request handling is disabled. Set Accept-Requests to 0 only if the feature you are planning to use is the MN-HA Shared Key Distribution support. If set to 1 (the default), FA User Authentication request handling is enabled.
HA-Address-Mismatch	<p>Specifies the action to take if an FA User Authentication request contains a specific HA-Address that is different than the HA-Address in the return list:</p> <ul style="list-style-type: none"> If set to response (the default), the HA-Address returned in the Access-Accept is the value from the mobile user's return list. If set to request, the HA-Address from the request is returned in the Access-Accept, overriding the HA-Address in the return list. If set to reject, the request is rejected.
Filter	<p>The name of the filter applied to attributes in an Access-Accept that is issued in response to an FA User Authentication request. The filter must be defined in the <code>filter.ini</code> file.</p> <p>If no filter is specified, all attributes are returned unchanged.</p>
HA-Address-Round-Robin-Group	<p>Enable Dynamic HA assignment and specifies the name of the round robin file used to control the assignment of HAs:</p> <p>For more information about Dynamic HA assignment and round robin files, see "Dynamic Home Agent" on page 29.</p>

Example

```
[FA-User-Auth-Requests]
Accept-Requests = 1
HA-Address-Mismatch = response
Filter = Filter1
HA-Address-Round-Robin-Group=ha_assign.rr
```

[HA-Key-Distribution-Requests] Section

The [HA-Key-Distribution-Requests] section (Table 7) specifies how HA Key Distribution requests should be handled.

Table 7. [HA-Key-Distribution-Requests] Section

Setting	Description
Accept-Requests	<ul style="list-style-type: none"> If set to 0, HA Key Distribution request handling is disabled. You should use this setting only if you are not expecting any FA User Authentication requests that ask for IPsec keying information. If set to 1 (the default), HA Key Distribution request handling is enabled.

Table 7. [HA-Key-Distribution-Requests] Section (Continued)

Setting	Description
Use-S-Request-As-Marker	<ul style="list-style-type: none"> If set to 1, the presence of the S-Request attribute is required for a request to be processed as an HA Key Distribution request. If set to 0 (the default), the only requirements an HA Key Distribution request must meet is that it not contain attributes that classify the request as an FA or MN-HA Shared Secret request and that the User-Name attribute consist of two ASCII-formatted IP addresses, with the second being a valid HA address.
Check-Source-Address	<ul style="list-style-type: none"> If set to 1, each HA Key Distribution request is validated to ensure that the HA address matches the source IP address of the Access-Request. If set to 0 (the default), this check is not performed.
Check-NAS-IP-Address	<ul style="list-style-type: none"> If set to 1, each HA Key Distribution request is validated to ensure that the HA address matches the value of the <code>NAS-IP-Address</code> attribute in the request. If set to 0 (the default), this check is not performed.
Auth	<ul style="list-style-type: none"> If set to native, each HA Key Distribution request will be authenticated against the password configured for a Native User in the Steel-Belted Radius server's database. The name of the Native User account that will be used for the check is constructed by concatenating the HA-Prefix value with an ASCII representation of the HA's address. If set to default (the default), the password check will be performed based on the definition of each HA in the [HAs] section.
Auth-Prefix	Set to the prefix string to which to append an HA address when HA Key Distribution requests are to be authenticated and the password for each HA is maintained in the server's database under a Native User account. The default string value is HA-.
Filter	<p>The name of the filter applied to attributes in an Access-Accept that is issued in response to an HA Key Distribution request. The filter must be defined in the <code>filter.ini</code> file.</p> <p>If no filter is specified, all attributes will be returned unchanged.</p>

Example

```
[HA-Key-Distribution-Requests]
Accept-Requests = 1
Filter = Filter2
Check-Source-Address = 1
Check-NAS-IP-Address = 0
Auth = default
```

[MN-HA-Shared-Key-Requests] Section

The [MN-HA-Shared-Key-Requests] section (Table 8) specifies how MN-HA Shared Key requests should be handled.

Table 8. [MN-HA-Shared-Key-Requests] Section

Setting	Description
Accept-Requests	<ul style="list-style-type: none"> If set to 0 (the default), local MN-HA Shared Key Distribution request handling is disabled. These requests can still be handled by proxy targets. If set to 1, MN-HA Shared Key Distribution request handling is enabled.
Filter	<p>The name of the filter applied to attributes in an Access-Accept that is issued in response to an MN-HA Shared Key Distribution request. The filter must be defined in the <code>filter.ini</code> file.</p> <p>If no filter is specified, all attributes will be returned unchanged.</p>

Example

```
[MN-HA-Shared-Key-Requests]
Accept-Requests = 1
Filter = Filter3
```

[HA-User-Auth-Requests] Section

The [HA-User-Auth-Requests] section (Table 9) specifies how HA User Authentication requests should be handled.

Table 9. [HA-User-Auth-Requests] Section

Setting	Description
Accept-Requests	<ul style="list-style-type: none"> If set to 0, HA User Authentication request handling is disabled, and any HA User Authentication requests are treated as being of type Other. If set to 1 (the default), HA User Authentication request handling is enabled.
HA-Address	<p>Specifies how an HA User Authentication request is to be recognized. Since this type of request is not required to contain a unique attribute, a mechanism must be specified to recognizing that this is a request from an HA that does not fit the criteria for HA Key Distribution or MN-HA Shared Key requests.</p> <ul style="list-style-type: none"> If set to source-IP-address (the default), the source address of the packet that contained the request must match one of the HA addresses listed in the [HAs] section. If set to NAS-IP-Address, the request must include a NAS-IP-Address attribute and the contents of this attribute must match one of the HA addresses listed in the [HAs] section.
Filter	<p>The name of the filter applied to attributes in an Access-Accept that is issued in response to an HA User Authentication request. The filter must be defined in the <code>filter.ini</code> file.</p> <p>If no filter is specified, all attributes will be returned unchanged.</p>

Example

```
[HA-User-Auth-Requests]
Accept-Requests = 1
HA-Address = source-IP-address
Filter = Filter4
```

[SIP-User-Auth-Requests] Section

The [SIP-User-Auth-Requests] section (Table 10) specifies how SIP User Authentication requests should be handled.

Table 10. [SIP-User-Auth-Requests] Section

Setting	Description
Accept-Requests	<ul style="list-style-type: none"> If set to 0, SIP User Authentication request handling is disabled, and any SIP User Authentication requests are treated as being of type Other. If set to 1 (the default), SIP User Authentication request handling is enabled.
Filter	<p>The name of the filter applied to attributes in an Access-Accept that is issued in response to a SIP User Authentication request. The filter must be defined in the <code>filter.ini</code> file.</p> <p>If no filter is specified, all attributes are returned unchanged.</p>

Example

```
[SIP-User-Auth-Requests]
Accept-Requests = 1
Filter = Filter5
```

[Other-Requests] Section

The [Other-Requests] section (Table 11) specifies how other requests (ones that do not fit in any of the other categories) should be handled.

Table 11. [Other-Requests] Section

Setting	Description
Filter	<p>The name of the filter to be applied to attributes in an Access-Accept in response to all requests of type Other. The filter must be defined in the <code>filter.ini</code> file.</p> <p>If no filter is specified, all attributes are returned unchanged.</p>

Example

```
[Other-Requests]
Filter = DefaultFilter
```

[Attributes] Section

The [Attributes] section lists the names of the special-purpose attributes used for Mobile IP. The [Attributes] section should read as follows.

```
[Attributes]
Pre-Shared-Secret-Request = 3GPP2-Pre-Shared-Secret-Request
Pre-Shared-Secret = 3GPP2-Pre-Shared-Secret
HA-Address = 3GPP2-Home-Agent-Address
Key-ID = 3GPP2-Key-ID
S-Key = 3GPP2-S-Key
S-Lifetime = 3GPP2-S-Lifetime
Correlation-Id = 3GPP2-Correlation-ID
Session-Continue = 3GPP2-Session-Continue
S-Request = 3GPP2-S-Request
MN-HA-SPI = 3GPP2-MN-HA-SPI
MN-HA-Shared-Key = 3GPP2-MN-HA-Shared-Key
```

NOTE: These are the standard 3GPP2 attributes specified in IS-835-B. You should not need to change anything in this section unless your FA or HA requires attributes that are not consistent with this standard.

[HAs] Section

The [HAs] section lists the address of each HA from which the server is willing to accept an HA Key Distribution request, along with an optional password.

```
[HAs]
HA-address [= password]
...
```

If the Auth setting in the [HA-Key-Distribution-Requests] section is set to native, only the *HA-address* needs to be specified, as all HA requests are checked against Native User accounts in the Steel-Belted Radius server's database. The remainder of this section describes options that apply only if HA-Auth is omitted or specified as default.

If the *password* parameter is omitted, no password check is performed. If the equal sign (=) is present but no password is specified, a check is performed for an empty (zero-length) password.

In the following example (with Auth set to default), HA Key Distribution requests from the HA at 200.200.200.1 are checked against the password swordfish, those from 200.200.200.2 are checked for an empty password, and those from 200.200.200.3 are accepted without performing a password check.

```
[HAs]
200.200.200.1 = swordfish
200.200.200.2 =
200.200.200.3
```

[FA-User-Auth-Requests/name] Sections

Multiple sections with names of the style [FA-User-Auth-Requests/*name*] may also exist in the 3gpp2.ini file. These sections are only referenced if a proxy realm's

configuration file (.pro) contains an FA-User-Auth-Requests-Section setting in its [3gpp2] section.

Specifying this option in a realm's configuration file puts the options in the matching FA User Authentication request processing section of the 3gpp2.ini file in effect for all FA User Authentication request transactions that are processed by the proxy realm. As a result, the settings in this section are used instead of the settings in the [FA-User-Auth-Requests] section for transactions processed against the proxy realm.

The options in these sections are identical to those documented for the [FA-User-Auth-Requests] section, which is described on [page 23](#).

Example

```
[FA-User-Auth-Requests/Acme]
Accept-Requests = 1
HA-Address-Mismatch = reject
Filter = FilterAcme
HA-Address-Round-Robin-Group=ha_assign.rr
```

Chapter 3

Advanced 3GPP2 Features

The Mobile IP module supports several advanced 3GPP2 features. To take advantage of these features, some configuration is necessary, as described in this chapter.

The mobility-related features are:

- Dynamic Home Agent

- Inter-PDSN Handoff

The subscriber management-related features are:

- New Session Hotlining

- Prepaid Data Services

- Dynamic Mobile Update

- LDAP Configuration Interface Extensions

- Support for HRPD Access Networks

Dynamic Home Agent

Dynamic Home Agent (HA) assignment is a method that Steel-Belted Radius uses to assign a Home Agent to a mobile IP session. Dynamic HA assignment facilitates load balancing among HAs because the assignment of HAs is shared among a pool of HAs.

Dynamic HA assignment makes use of the Steel-Belted Radius feature known as Round-Robin Load Balancing. (See the *Steel-Belted Radius Administration Guide* and *Steel-Belted Radius Reference Guide* for details about Round Robin Load Balancing.)

Configuring Dynamic Home Agent

Steel-Belted Radius assigns an HA to a session based on various configuration choices. The HA may be a specific HA in the configured return list (static assignment), assigned from a round robin pool (dynamic assignment), or a specific HA requested in the access-request.

Steel-Belted Radius uses dynamic HA Assignment when you have configured it for this method of assignment.

To configure dynamic home agent:

- 1 In the `3gpp2.ini` file [Settings] section, make sure the following line appears:

```
Enable=1
```

- 2 Place the following line in the `3gpp2.ini` file [FA-User-Auth-Requests] section:

```
HA-Address-Round-Robin-Group = round-robin-filename
```

where:

`round-robin-filename` is the filename of the `.rr` file used to control the assignment of HAs.

The special round robin group (HA-Address-Round-Robin-Group) can contain only round robin filenames used for the purpose of dynamic home agent assignment.

- 3 Create a round robin (`.rr`) file to control the assignment of HAs. See the *Steel-Belted Radius Administration Guide* for instructions for creating round robin files.

Example

In the following example, dynamic HA assignment is enabled by the specification of an `.rr` group in the [FA-User-Auth-Requests] section. The HA addresses are specified in the `.rr` file called `ha_assign.rr`. Addresses are assigned according to a ratio. The first address is assigned 20/100 of the time. The second address is assigned 40/100 of the time. The third address is assigned 30/100 of the time.

NOTE: The values in the [Sets] section do not have to add up to 100, as shown in the example.

3gpp2.ini File

```
[Settings]
Enable = 1
[FA-User-Auth-Requests]
HA-Address-Round-Robin-Group = ha_assign.rr
```

ha_assign.rr File

```
[Sets]
Set1=20
Set2=50
Set3=30
[Set1]
3GPP2-Home-Agent-Address = 10.10.10.2
[Set2]
3GPP2-Home-Agent-Address = 10.10.10.3
[Set3]
3GPP2-Home-Agent-Address = 10.10.10.4
```

Inter-PDSN Handoff

Mobile sessions may need to be transferred from one PDSN (Packet Data-Serving Node) to another. Steel-Belted Radius can detect that a session exists for the original PDSN, but is now being transferred to a different PDSN because of the user's mobility. In this case, Steel-Belted Radius assigns the same HA address and Home Address that the session originally used so that the “inter-PDSN handoff” is seamless to the subscriber.

Inter-PDSN Handoff Process

The Inter-PDSN Handoff process is described in the following list. [Table 12](#) and [Table 13 on page 32](#) summarize Inter-PDSN Handoff assignment of Home Agent address and Home Address.

- 1 Steel-Belted Radius determines if an Access-Request is new or “handed off” in the following way:
 - a If the User-Name attribute or the Calling-Station-ID (or both) match that of an existing session, the Access-Request is determined to be an inter-PDSN handoff. The User-Name attribute contains the Network Access Identifier (NAI) value, and the Calling-Station-ID attribute contains the IMSI (International Mobile Subscriber Identity) value.
 - b The use of User-Name or Calling-Station-ID (or both) as identifiers of an existing session is configured in the `UniqueDeviceIdentifier` field in the `3gpp2.ini` file.

NOTE: See [“Configuring Inter-PDSN Handoff” on page 32](#) for more information.
- 2 Steel-Belted Radius determines the appropriate Home Agent address to send to the handed-off session in the Access-Accept (as the value of 3GPP2-Home-Agent Address):
 - a If the Access-Request contains a `3gpp2-Home-Agent-Address` value of 0.0.0.0. or 255.255.255.255 for the Home Agent address, then the existing Home Agent address already in use for the session is assigned.
 - b If the Access-Request contains the `3gpp2-Home-Agent-Address` value of the specific Home Agent address already in use for the session, then the existing Home Agent address is assigned.
 - c If the Access-Request contains a specific `3gpp2-Home-Agent-Address` value that is different from the Home Agent address already in use for the session, then the access-request is rejected.
- 3 Steel-Belted Radius determines the appropriate Home Address to send to the handed-off session in the Access-Accept (as the value of Framed-IP-Address):
 - a If the Access-Request contains a `Framed-IP-Address` value of 0.0.0.0. or 255.255.255.255 for the Home Address, then the existing Home Address already in use for the session is assigned.

- b If the Access-Request contains a Framed-IP-Address value of the specific Home Address already in use for the session, then the existing Home Address is assigned.
- c If the Access-Request contains a specific Framed-IP-Address value that is different from the Home Address already in use for the session, then the access-request is rejected.

Inter-PDSN Handoff Assignment of Home Agent and Home Address

Table 12 and Table 13 summarize the factors that affect Home Agent and Home Address assignment for inter-PDSN handoff.

Table 12. Home Agent Assignment for Inter-PDSN Handoff

Home Agent (HA) Requested in the Access-Request	Home Agent (HA) Assigned
0.0.0.0 or 255.255.255.255	Same HA address as already in use for the session
Specific HA address already in use for the session	Same HA address as already in use for the session
Specific HA address requested but differs from the HA address of the existing session	Access rejected

Table 13. Home Address Assignment of Inter-PDSN Handoff

Home Address Requested in the Access-Request	Home Address Assigned
0.0.0.0 or 255.255.255.255	Same Home Address as already in use for the session
Specific Home Address already in use for the session	Same Home Address as already in use for the session
Specific Home Address requested but differs from the Home address of the existing session	Access rejected

Configuring Inter-PDSN Handoff

You must configure the `3gpp2.ini` file and the `radius.ini` file to enable inter-PDSN handoff. If inter-PDSN handoff is not enabled, the Home Agent address and Home address will be allocated as if the access-request is for a new session.

To configure inter-PDSN handoff:

- 1 In the [Settings] section of `3gpp2.ini`, make sure `enable=1` to enable 3gpp2 processing.
- 2 In the [Settings] section of `3gpp2.ini`, make sure the following line appears (to enable inter-PDSN handoff):

```
MIPSessionsForDeviceMustHaveSameHomeAndHAAddresses = 1
```

- 3 In the [Settings] section of `3gpp2.ini`, complete the `UniqueDeviceIdentifier` field to determine how the session will be recognized as an existing session to be handed off:

```
UniqueDeviceIdentifier=[User-Name] [, Calling-Station-Id]
```

where:

the value to the right of the equals sign may be `User-Name` or `Calling-Station-Id` or `User-Name, Calling-Station-Id`.

- 4 In the `radius.ini` file [Configuration] section, include the line:

```
FramedIpAddressHint=check-pool
```

NOTE: *FramedIpAddressHint must be set to check-pool (not to yes or no).*

Example

`3gpp2.ini` File

```
[Settings]
enable = 1
MIPSessionsForDeviceMustHaveSameHomeAndHAAAddresses = 1
UniqueDeviceIdentifier=User-Name, Calling-Station-Id
```

`radius.ini` file

```
[Configuration]
FramedIpAddressHint=check-pool
```

New Session Hotlining

The Mobile IP module supports new session hotlining (NSHL) for 3GPP2. NSHL allows you to divert subscribers who are initiating new sessions to an alternate destination (web site or IP address). For example, subscribers might be diverted to websites containing notification of a delinquent account, information about expiring accounts, or advertisements about new subscription packages.

Configuring New Session Hotlining

You must configure Steel-Belted Radius and the Mobile IP module to recognize that NSHL is supported and initiated. Configuration for NSHL involves the following activities:

- ▶ Enabling new session hotlining
- ▶ Configuring the `nshl.att` file
- ▶ Creating hotline profiles
- ▶ Assigning hotline profiles to users:
 - ▷ Assigning hotline profiles with the SQL database
 - Populating the SQL Database
 - Configuring the `radsql.aut` file or `radsqljdbc.aut` file
 - ▷ Assigning hotline profiles with Steel-Belted Radius

Each of these configuration activities is described in the following sections.

Enabling New Session Hotlining

New session hotlining must be enabled in the `3gpp2.ini` file and the `radius.ini` file.

To enable new session hotlining:

- 1 In the [Settings] section of `3gpp2.ini`, set `enable` to 1.
- 2 Add the following section to the `radius.ini`:


```
[AttributeEditing]
NSHL
```
- 3 Configure the `nshl.att` file as described in “Configuring the `nshl.att` File” on page 34.

Example

`3gpp2.ini` File

```
[Settings]
Enable = 1
```

`radius.ini` File

```
[AttributeEditing]
NSHL
```

NOTE: To disable new session hotlining, comment out the `NSHL` line in `radius.ini` and set `enable=0` line in the [Bootstrap] section of the `nshl.att` file.

Configuring the `nshl.att` File

The `nshl.att` file must contain certain lines to enable new session hotlining.

NOTE: The [Settings] section of the `nshl.att` file can be updated by a HUP signal.

To configure `nshl.att` for new session hotlining:

- 1 In the [Bootstrap] section, make the following changes:
 - a Change `Enable=0` to `Enable=1`.
 - b Make sure the following lines are in the file:


```
LibraryName=nshl.so
InitializationString=NSHL
```
- 2 In the [Settings] section, optionally set `ConfigLog` as follows:
 - ▷ `ConfigLog=ConsoleAndLog` sends the log information to both the console and the log.
 - ▷ `ConfigLog=Console` sends the log information to the console only.
 - ▷ `ConfigLog=Log` sends the log information to the log only.
- 3 In the [Settings] section, set the Hotline-Device value to PDSN, HA, Both, or None, depending on the type of device. This line ensures that the appropriate

hotline attributes are returned in the Funk-3GPP2-Hot-Line-Profile attribute as follows:

Table 14. Results of Hotline-Device Settings

Hotline-Device Setting	Type of Authentication	Result Returned in Funk-3GPP2-Hot-Line-Profile Attribute
PDSN	FA User or SIP User	Hotlining attributes are returned.
PDSN	Any type other than FA User or SIP User	No hotlining attributes are returned.
HA	HA User	Hotlining attributes are returned.
HA	Any type other than HA User	No hotlining attributes are returned.
Both	HA User or FA User or SIP User	Hotlining attributes are always returned.
None	N/A	No hotlining attributes are returned.

NOTE: The None setting ensures that no Funk-3GPP2-Hot-Line-Profile attribute is returned.

Example

nshl.att File

```
[Bootstrap]
enable=1
InitializationString=NSHL
LibraryName=nshl.so
[Settings]
Hotline-Device=PDSN
ConfigLog = ConsoleAndLog
```

Hotlining Prepaid Sessions

You may choose to hotline prepaid sessions. For more information about the Prepaid module, see “Prepaid Data Services” on page 41.

To configure the Mobile IP module to allow hotlining of prepaid sessions:

- 1 In the [AttributeEditing] section of `radius.ini`, add the word `prepaidAttr`. (This line must precede the line for `NSHL` in the [AttributeEditing] section.)
- 2 Make sure the word `NSHL` appears in the [AttributeEditing] section of `radius.ini`, to allow new session hotlining functionality. (This line must come after the line for `prepaidAttr` in the [AttributeEditing] section.)
- 3 In the [Settings] section of the `nshl.att` file, add the following line:
`AllowHotliningOfPrepaidSession=1`

Example

radius.ini file

```
[AttributeEditing]
prepaidAttr
NSHL
```

nshl.att file

```
[Settings]
AllowHotliningOfPrepaidSession=1
```

Disabling New Session Hotlining

To disable new session hotlining:

- 1 Comment out the NSHL line in `radius.ini`.
- 2 Set `Enable=0` in the [Bootstrap] section of the `nshl.att` file.

Creating Hotline Profiles

Steel-Belted Radius must be configured to include hotline profiles. The profiles specify the specific website where hotlined sessions are to be directed. For example, you might create a profile called “DELINQUENT” that diverts subscribers with delinquent accounts to a website where they can pay their fees. You might also create a profile called “RENEW” to divert subscribers with expiring subscriptions to a website where they can renew their subscriptions.

The process of creating profiles is described in the *Steel-Belted Radius Administration Guide*. The following process describes the steps involved for creating profiles to support new session hotlining.

To configure Steel-Belted Radius for new session hotlining:

- 1 Add a profile name to Steel-Belted Radius for each new session hotlining profile you want to create.

For more information about creating Steel-Belted Radius profiles, see the *Steel-Belted Radius Administration Guide*.
- 2 Add one or more of the new session hotlining attributes and values to each profile.

Table 15. New Session Hotlining Attributes

New Session Hotlining Attribute	Description
3GPP2-Hot-Line-Profile-Id	<p>Invokes a set of pre-configured hotlining rules configured at the hotline device (PDSN or HA). This attribute is used if the hotline device is set up to invoke a set of rules for hotlining, rather than sending the information as values for the hotline attributes described in the remainder of this table.</p> <p>For example, the hotline device may be set up so that if 3GPP2-Hot-Line-Profile-Id is sent in the Access-Accept with a value of DELINQUENT, the user is redirected to <code>http://www.pay_now.com</code>.</p> <p>If the 3GPP2-Hot-Line-Profile-Id attribute is used, the hotlining rules are set up at the hotline device and there is probably no reason to use the other attributes listed in this table. On the other hand, if the other attributes in this table are used to specify hotlining rules, there is probably no reason to use 3GPP2-Hot-Line-Profile-Id.</p>

Table 15. New Session Hotlining Attributes (Continued)

New Session Hotlining Attribute	Description
3GPP2-Hot-Line-Accounting-Ind	Indicates reason the subscriber has been hotlined.
3GPP2-Filter-Rule	Sets up packet filter rules that permit traffic.
3GPP2-HTTP-Redirection-Rule	Instructs the hotlining device where to redirect HTTP flows (web address).
3GPP2-IP-Redirection-Rule	Instructs the hotlining device where to redirect packet flows (IP address).

NOTE: For more information about 3GPP2 hotlining attributes, see *cdma2000 Wireless IP Network Standard: Accounting Services and 3GPP2 RADIUS VSAs* which are available at <http://www.3gpp2.org>.

Hotline Capability Attribute

The hotlining device includes the attribute 3GPP2-Hot-Line-Capability to indicate its hotlining capabilities. For example, the device might support the HTTP redirection rule but not the IP redirection rule. Only the hotlining rules that the device can support are returned.

Filtering Hotlining Capabilities

You can exclude certain hotlining attributes even though a hotlining device has the capability to support them. For example, a hotlining device might support both the HTTP redirection rule and the IP redirection rule, but you may choose to exclude the IP redirection rule.

NOTE: It is more likely that you will choose to include and exclude hotlining attributes within the profile definitions than with the filtering capability. See “[Creating Hotline Profiles](#)” on page 36.

To filter a rule, enter an Exclude statement in the [FAFilter] or [HAFilter] section of the `filter.ini` file. For example:

```
[FAFilter]
Allow
Exclude 3GPP2-IP-Redirection-Rule
```

In the [FA-User-Auth-Requests] section or the [SIP-User-Auth-Requests] section of the `3gpp2.ini` file, if a filter is specified, make sure that the filter is defined in the `filter.ini` file and that the filter allows (or does not exclude) all the attributes relevant to the new session hotlining module. These attributes are:

```
3GPP2-Hot-Line-Accounting-Ind
3GPP2-Hot-Line-Profile-Id
3GPP2-Filter-Rule
3GPP2-HTTP-Redirection-Rule
3GPP2-IP-Redirection-Rule
```

NOTE: A filter that is undefined in `filter.ini` defaults to rejecting all attributes.

For more information about filtering and the `filter.ini` file, see the *Steel-Belted Radius Reference Guide*.

Assigning Hotline Profiles to Users

Hotline profiles must be assigned to specific users as appropriate. For example, you may have created a profile called DELINQUENT to redirect users who have late accounts to a web site where they can pay their bills.

There are two ways that you can assign profiles to users:

- ▶ Assign profiles with the SQL database (recommended).
With this method, you associate a user with a hotline profile in the SQL database. This method is recommended because it can be automated and is more practical for assigning profiles to large numbers of users at once.
- ▶ Assign profiles with Steel-Belted Radius.
With this method, you associate a user with a hotline profile using the Steel-Belted Radius return list.

Figure 2 illustrates that NSHL profiles can be created to redirect (“hotline”) users to specific websites before they are granted network access. The appropriate profile is assigned to a specific user either (a) by populating the SQL database or (b) by entering profile names and user names in Steel-Belted Radius.

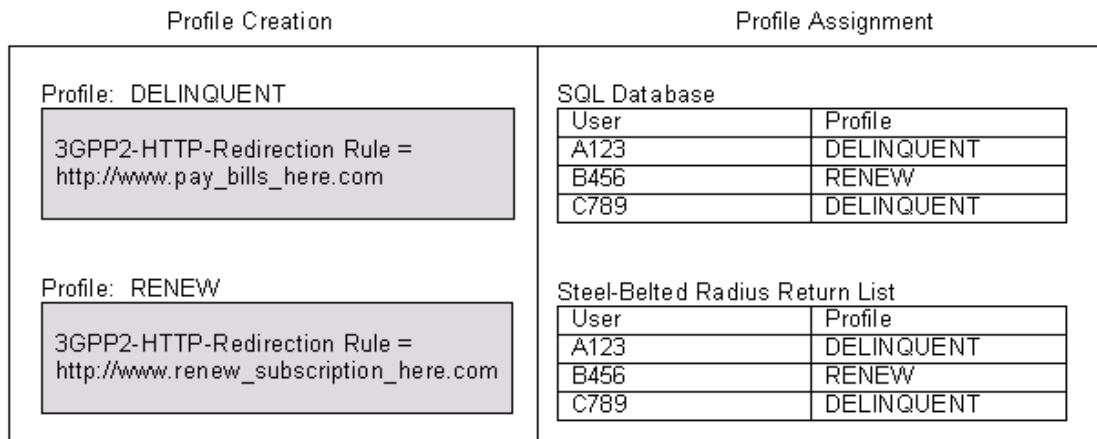


Figure 2 Methods for Assigning Profiles to Users

Each of these methods for assigning profiles to users is described in the following sections.

Assigning Hotline Profiles with the SQL Database

To assign profiles to users with the SQL database, you must complete the following two activities:

- ▶ Populate the SQL database with profile names
- ▶ Configure the `radsq1.aut` file or `radsq1jdbc.aut` file

Populating the SQL Database with Profile Names

The SQL database must contain a column for the profile name. The column must be populated with the appropriate profile name to be used for each subscriber (or left blank if no profile is to be assigned).

Subscriber_ID	Profile_Name
A1000	Delinquent
B2000	Renew
C3000	Delinquent
D4000	Delinquent
E5000	
F6000	Renew

Figure 3 SQL database initialized with profiles to be used for each subscriber

In [Figure 3](#), the profile name Delinquent is placed in the rows corresponding to delinquent subscribers. The profile name Renew is placed in the rows corresponding to subscribers due for renewal. Note that the profiles Delinquent and Renew must have been created using Steel-Belted Radius, as described in [“Creating Hotline Profiles” on page 36](#).

The profiles contain the address where subscribers will be diverted (“hotlined”). For example, subscriber A1000 will be hotlined to a website for delinquent subscribers. The address of this website is contained in the profile called “Delinquent” created within Steel-Belted Radius. Subscriber E5000 contains no profile name. Therefore, this subscriber will connect directly to the network without first being hotlined to another website.

To configure the SQL database for new session hotlining:

- 1 In the SQL database, add a column for the hotline profile names. The column header is specified in the SQL SELECT statement, as described in [“Configuring the radsql.aut or radsqldb.aut File” on page 39](#).
- 2 For each subscriber, enter the profile name. Leave the column blank if the subscriber is not to be hotlined, as shown in [Figure 3](#).

Configuring the radsql.aut or radsqldb.aut File

The `radsql.aut` file or `radsqldb.aut` file controls SQL authentication of the SQL database. The SQL database contains the hotline profile name (if any) to be used for each subscriber.

The following procedure provides the basic information for configuring `radsql.aut` file or `radsqldb.aut` file for new session hotlining.

To configure the `radsql.aut` file or `radsqldb.aut` file for new session hotlining:

- 1 Place the following line in the [Results] section:

```
Funk-3GPP2-Hot-Line-Profile=n/m
```

where:

n is the location in the SQL SELECT statement that corresponds to the column heading of the SQL database. For example, if n is 1, then the first position in the SQL SELECT statement contains the name of the column that corresponds to the profile name.

m is the maximum number of characters in that column.

For example: Funk-3GPP2-Hot-Line-Profile=1/48

- 2 In the [Query] section, place a SQL statement that queries the database for the Funk-3GPP2-Hot-Line-Profile variable (the name of the profile to be used for the subscriber).
- 3 In the [Settings] section, place a Connect statement to specify the string that must be passed to the database engine to establish a connection to the database.

SQL authentication is discussed in depth in the *Steel-Belted Radius Administration Guide*. For syntax descriptions of lines within the `radsq1.aut` file or `radsq1jdbc.aut` file, see the *Steel-Belted Radius Reference Guide*.

Example

In the following example, the [Results] section indicates that the heading of the database column used for the profile is in the first position in the SQL statement, because the 1 in 1/48 indicates position 1 in the SQL statement. The SQL statement uses the variable name `Profile_Name` in the first position. Therefore, the value of Funk-3GPP2-Hot-Line-Profile is contained in the SQL database in the column with the heading `Profile_Name`.

`radsq1.aut` file File

```
[Settings]
Connect = Connect=databasename/databasepassword
[Query]
SQL = SELECT Profile_Name, FROM SQL_databasename WHERE
Subscriber_ID = @NAI
[Results]
Funk-3GPP2-Hot-Line-Profile = 1/48
```

Assigning Hotline Profiles with Steel-Belted Radius

To assign profiles to users with Steel-Belted Radius, you select a user and then add the Funk-3GPP2-Hot-Line-Profile attribute and its associated profile name.

To assign profiles to users with the Steel-Belted Radius return list:

- 1 Run Steel-Belted Radius Administrator and log into the appropriate Steel-Belted Radius server.

2 Click **Users**.

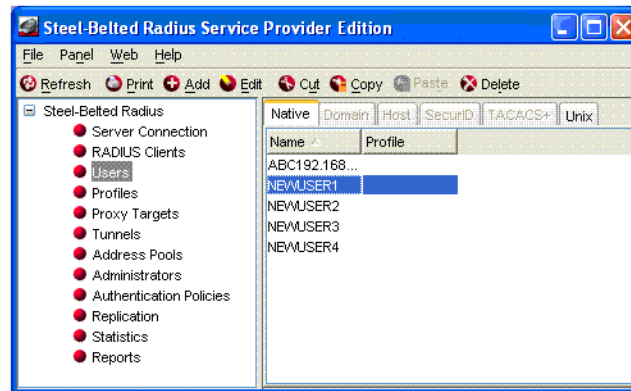


Figure 4 Display of Users to Which Profiles Can be Assigned

- 3 Select a user and click **Edit**.
- 4 Click the **Return List** tab if it is not already active.
- 5 Click **Add**.
- 6 Select Funk-3GPP2-Hot-Line-Profile in the **Attributes** list.
- 7 In the **String** field, type the name of the hotlining profile to be assigned to the user.

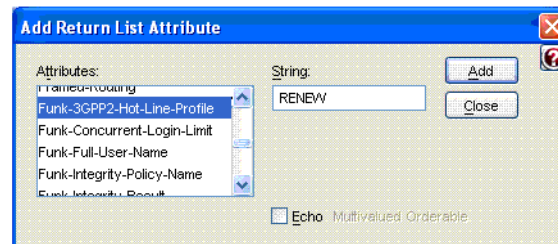


Figure 5 Profile Name Assigned to Funk-3GPP2-Hot-Line-Profile Attribute

Prepaid Data Services

The Mobile IP module supports prepaid data services for 3GPP2 FA user Authentication Requests and SIP User Requests. (See “FA User Authentication Requests” on page 11 and “SIP User Authentication Requests” on page 6.)

NOTE: The Prepaid module is designed to comply with the prepaid specifications defined in *cdma2000 Wireless IP Network Standard, X.P0011-006-C, PrePaid Packet Data Services, Accounting Services and 3GPP2 RADIUS VSAs. v0.5*, available at <http://www.3gpp2.org> and with the *Parlay 5.0 Specifications (Part 12)* available at <http://www.parlay.org>.

Overview

Support for prepaid data services includes the following features:

- ▶ Recognition of prepaid attributes and subtypes

- ▶ Initial authentication and online authorization of prepaid sessions
- ▶ Support for volume-based or duration-based prepaid services
- ▶ Reconciliation of prepaid accounts when inter-PDSN handoff occurs
- ▶ Communication with a third-party billing server which manages charging sessions, reserves amounts from the subscriber accounts, and debits subscriber accounts.
- ▶ Relaying of subscriber account balance and usage information to the PDSN.

Components of the Prepaid Module

The Prepaid module works in conjunction with Steel-Belted Radius to interface with a prepaid server. The Prepaid module carries prepaid account information to and from the prepaid server. The prepaid server is responsible for its own quota management function, including maintaining the user, quota, balance, and allocation information. The Mobile IP Prepaid module, working with Steel-Belted Radius, is responsible for relaying prepaid user account/session information back to the PDSN.

Steel-Belted Radius uses three plugin libraries to pass accepted access requests for prepaid service to a third-party prepaid billing server using a Parlay interface. Plugin libraries work with Steel-Belted Radius to accomplish additional functionality. Parlay is a protocol used for communicating with an external prepaid server. The Mobile IP module makes use of the Parlay interface as a means of communicating with the billing server.

The three plugin programs are:

PrepaidAcct.acct — This plugin handles the timeout (closing) of outstanding prepaid sessions and ensures that the appropriate sessions are closed when an account-stop message is received.

PrepaidAttr.att — Controls the authorization, update, and closing of prepaid sessions.

ParlayPPSPlugin.gen — Manages the connection with the billing server using the Parlay interface.

Figure 6 illustrates the relationship of the components of prepaid billing.

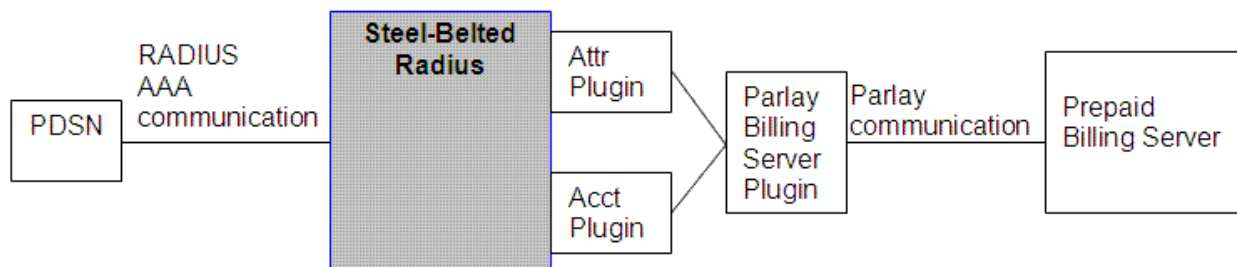


Figure 6 Components of Prepaid Billing Module

The third-party billing server is responsible for keeping track of the individual usage of subscribers and assigning duration or volume allocations to users.

When an Access-Request is sent to Steel-Belted Radius with prepaid attributes attached, prepaid data services are initiated for the subscriber, and the session information is passed through the Parlay interface to the prepaid service provider for handling.

Configuring Prepaid Functionality

You must configure Steel-Belted Radius and the Mobile IP module to recognize that prepaid functionality is supported and initiated. Configuration for prepaid involves the following activities:

- ▶ [Configuring the 3gpp2.ini File for the Prepaid Module \(page 43\)](#)
- ▶ [Configuring Filters for Prepaid Attributes \(page 44\)](#)
- ▶ [Configuring the prepaidAcct.acc Plugin File \(page 45\)](#)
- ▶ [Configuring the prepaidAttr.att Plugin File \(page 45\)](#)
- ▶ [Configuring the parlayPPSplugin.gen File \(page 47\)](#)
- ▶ [Configuring the radius.ini File for the Prepaid Module \(page 50\)](#)
- ▶ [Sending and Receiving Prepaid Attributes \(page 50\)](#)
- ▶ [Configuring Prepaid Timeouts \(page 54\)](#)
- ▶ [Using New Session Hotlining and Prepaid Sessions Together \(page 55\)](#)

Each of these configuration activities is described in the following sections.

Configuring the 3gpp2.ini File for the Prepaid Module

Prepaid functionality is a 3GPP2 feature. You must configure the `3gpp2.ini` file for 3GPP2 features. The `3gpp2.ini` file is included with the Mobile IP module.

NOTE: *The `3gpp2.ini` file is read whenever Steel-Belted Radius restarts or receives a HUP signal. However, the [Settings] section cannot be updated via a HUP signal.*

To configure the `3gpp2.ini` file:

- 1 In the `3gpp2.ini` file [Settings] section, set `enable=1`.
- 2 Make sure that the `3gpp2.ini` file includes an [FA-User-Auth-Requests] section or a [SIP-User-Auth-Requests] section.

One of these sections is required because Prepaid module features are supported only for FA User Authentication requests and SIP User Authentication requests.

- 3 Make sure that the `3gpp2.ini` file includes an [Authorize-Only-Requests] section. This section must include the line `Accept-Requests=1`.
- 4 If you specify a filter in the [FA-User-Auth-Requests], [SIP-User-Auth-Requests], or [Authorize-Only-Requests] section, make sure that the filter is defined in the `filter.ini` file as described in [Configuring Filters for Prepaid Attributes](#).

Configuring Filters for Prepaid Attributes

Filters allow you to modify the returned attributes by request type. You can specify attributes that should be allowed in the Access-Accept or specify certain attributes that should be excluded from the Access-Accept. If you specify a filter in the [FA-User-Auth-Requests], [SIP-User-Auth-Requests], or [Authorize-Only-Requests] sections of `3gpp2.ini`, make sure that the filter is defined in the `filter.ini` file.

FA and SIP filters are applied to initial requests. An initial request is the first, authenticating Access-Request. Authorize-Only filters are applied to online requests. Online requests are Access-Requests that tell the billing server to update account balance information and ask for more allocation of volume or duration.

For more information about Prepaid module attributes, see [“Sending and Receiving Prepaid Attributes” on page 50](#) and [“Filtering Prepaid Attributes” on page 53](#). For more information about filters, see the *Steel-Belted Radius Administration Guide*.

All three sections of `3gpp2.ini` that are related to Prepaid module functionality ([FA-User-Auth-Requests], [SIP-User-Auth-Requests], and [Authorize-Only-Requests]) must allow all the attributes related to online requests. These attributes are:

```
3GPP2-Prepaid-Acct-Quota
3GPP2-Session-Continue
Event-Timestamp
User-Name
Calling-Station-ID
Service-Type
3GPP2-Correlation-ID
Message Authenticator
3GPP2-Service-Reference-ID
```

The [FA-User-Auth-Requests] and [SIP-User-Auth-Requests] sections must allow all the attributes listed above plus these two attributes:

```
3GPP2-Prepaid-Acct-Capability
3GPP2-Session-Term-Capability
```

NOTE: A filter that is specified in the `3gpp2.ini` file but undefined in `filter.ini`, defaults to rejecting all attributes.

NOTE: Attributes needed for RADIUS communication must also be allowed by the filters (not just the Prepaid module attributes). For more information about RADIUS attributes, see the *Steel-Belted Radius Administration Guide*.

Example

```
3gpp2.ini file
[Settings]
Enable = 1
[FA-User-Auth-Requests]
Accept-Requests = 1
Filter = FAFilter
[SIP-User-Auth-Requests]
Accept-Requests = 1
Filter = SIPFilter
[Authorize-Only-Requests]
Accept-Requests = 1
Filter = AuthorizeOnlyFilter
```

```

filter.ini file
[FAFilter]
Allow
Add Session-Timeout 36000
[SIPFilter]
Allow
[AuthorizeOnlyFilter]
Allow

```

NOTE: “Allow” allows all attributes in the request without excluding any attributes. “Add Session-Timeout” is described in “Configuring Prepaid Timeouts” on page 54.

Configuring the prepaidAcct.acc Plugin File

The `prepaidAcct.acc` file handles the timeout (closing) of outstanding prepaid sessions and ensures that the appropriate sessions are closed when an account-stop message is received. Accounting stops may be received by Steel-Belted Radius from the PDSN, or they may be internally generated for session timeouts, or they may be sent when sessions are terminated with the Steel-Belted Radius Administrator software.

The `prepaidAcct.acc` file is included with the Mobile IP module.

NOTE: The `prepaidAcct.acc` file requires that the `radius.ini` file has the correct configuration for `AcctAutoStopEnable`, `SessionTimeoutOnMissedAcctStopEnable`, and `SessionTimeoutOnMissedAcctStopGracePeriodSec`. See “Configuring the `radius.ini` File for the Prepaid Module” on page 50 for more information.

To configure the `prepaidAcct.acc` plugin file:

- 1 In the [Bootstrap] section, set `Enable` to 1.
- 2 Make sure that the following lines appear in the [Bootstrap] section:

```

LibraryName=prepaidAcct.so
InitializationString=PrepaidAcct

```

Example

```

prepaidAcct.acc file
[Bootstrap]
LibraryName=prepaidAcct.so
Enable=1
InitializationString=PrepaidAcct

```

Configuring the prepaidAttr.att Plugin File

The `prepaidAttr.att` file configures an attribute editing plugin file that communicates with `ParlayPPSPlugin`, which, in turn, communicates with the prepaid server using the Parlay interface.

The `prepaidAttr.att` file is included with the Mobile IP module.

To configure the `prepaidAttr.att` plugin file:

- 1 Configure the [Bootstrap] section, as described in Table 16 on page 46.
- 2 Configure the [Settings] section, as described in Table 17 on page 46.

NOTE: The `prepaidAttr.att` file is read whenever Steel-Belted Radius restarts or receives a HUP signal. Only the settings in the [Settings] section are reconfigured (not the settings in the [Bootstrap] section).

[Bootstrap] Section of prepaidAttr.att

The [Bootstrap] section (Table 16) enables Prepaid module functionality.

Table 16. [Bootstrap] Section

Setting	Description
Enable	<ul style="list-style-type: none"> Set to 1 to enable Prepaid module functionality. Set to 0 to disable Prepaid module functionality.
AcceptsAuthorizeOnly	Must be set to 1.
LibraryName	Must be set to <code>prepaidAttr.so</code>
InitializationString	Must be set to <code>PrepaidAttr</code>

[Settings] Section of prepaidAttr.att

The [Settings] section (Table 17) provides information that is needed by the Prepaid module.

Table 17. [Settings] Section

Setting	Description
PrepaidSubscriberIDAttribute	Set the value of PrepaidSubscriberID attribute to either User-Name or Calling-Station-ID. This setting defines which incoming attribute is used to identify the billing server. For example, if the billing server identifies users by their mobile number, use the attribute Calling-Station-ID.
SessionTerminationCapability	<p>Set the value of SessionTerminationCapability to the value to be returned in the initial Access-Accept for the 3GPP2-Session-Term-Capability attribute. Valid values are 0, 1, 2, and 3:</p> <ul style="list-style-type: none"> 0 - No session termination capability 1 - Dynamic-Authorization-Only 2 - Registration-Revocation-Only 3 - Dynamic-And-Reg-Revocation <p>The SessionTerminationCapability setting is used in case another device might send a Change of Authorization message to the PDSN.</p>

Table 17. [Settings] Section (Continued)

Setting	Description
ConfigLog	Specify the method for capturing Prepaid module configuration information. ConfigLog=None means that configuration information will not be captured. ConfigLog=ConsoleAndLog sends the log information to both the console and the log. ConfigLog=Console sends the log information to the console only. ConfigLog=Console sends the log information to the log only.
SessionTimeoutSeconds	Specify the number of seconds after the first Access-Accept after which the session automatically times out. This value is active only if the following settings are set to 1 in radius.ini: <ul style="list-style-type: none"> AcctAutoStopEnable SessionTimeoutOnMissedAcctStopEnable

Example

```

prepaidAttr.att file
[Bootstrap]
LibraryName=prepaidAttr.so
Enable=1
InitializationString=PrepaidAttr
AcceptsAuthorizeOnly=1

[Settings]
PrepaidSubscriberIDAttribute=User-Name
SessionTerminationCapability=1
ConfigLog=Log
SessionTimeoutSeconds=600

```

Configuring the parlayPPSplugin.gen File

The parlayPPSplugin.gen file configures the Parlay Prepaid Server plugin that sets up the connection between Steel-Belted Radius and the billing server using a Parlay interface.

The parlayPPSplugin.gen file is included with the Mobile IP module.

NOTE: The parlayPPSplugin.gen file is read whenever Steel-Belted Radius restarts or receives a HUP signal. Only the settings in the [Settings] section are reconfigured (not the settings in the [Bootstrap] section).

Table 18 defines the items that you must configure in the ParlayPPSPlugin.gen file.

Table 18. ParlayPPSPlugin.gen Configuration Variables

Configuration Variable	Description
------------------------	-------------

Table 18. *ParlayPPSPlugin.gen Configuration Variables (Continued)*

ChargingManagerLoc	CORBA URL of the <code>IpChargingManager</code> object that is the starting point for Parlay charging activities. Obtain this value from the billing service provider.
ChargingType	Indicate which type of prepaid service is to be used (volume-based prepaid service or duration-based prepaid service). The type is a single setting and applies to all transactions on the server. Valid values are: <ul style="list-style-type: none"> • Volume • Duration
MerchantName	Name of the network passed to the billing server when a prepaid session is initiated. Obtain this value from the billing service provider.
AccountID	Identifying number of the network passed to the billing server when a prepaid session is initiated. Obtain this value from the billing service provider.
ThresholdRatio	When the volume threshold or duration threshold is reached, Steel-Belted Radius must request additional quota to continue the session. The <code>ThresholdRatio</code> is used to calculate the volume threshold or duration threshold. See “Determining the Volume or Duration Threshold” on page 48 for more information about the <code>ThresholdRatio</code> and the threshold.
VolumeUnitMultiplier	Multiplier to be applied to volume amounts generated by the Parlay plugin to turn the volume amounts into bytes. <code>VolumeUnitMultiplier</code> is only needed if: Prepaid amounts are measured as volume rather than duration and The billing server does not express volume in bytes
ListSessionOnHUP	Valid values are: <ul style="list-style-type: none"> • True — When a HUP signal is detected, Steel-Belted Radius logs a list of the currently active Prepaid module sessions in the <code>ParlayPrepaidServer</code> section of the log file. • False — No Prepaid module session information is sent to the log.
RequestTimeoutMilliseconds	If a request from Steel-Belted Radius to the billing server for a debit or for more quota exceeds this amount of time (from request to response), the request times out and an <code>Access-Reject</code> is returned.

Determining the Volume or Duration Threshold

The volume or duration threshold is the point at which Steel-Belted Radius must request more allocation of volume or duration units for a session to continue.

Threshold is calculated as:

$$(\text{units used}) + (\text{ThresholdRatio} * \text{unused units})$$

For example, if 1000 units are initially allocated and the `ThresholdRatio` is .9, then the threshold is 900 calculated as $(0) + (.9 * 1000)$. Additional allocation must be made by the prepaid server after 900 units are used. At that time, 1000 units may be additionally

allocated. The threshold is now 1800 calculated as $(900) + (.9 * 1000)$. When the 1800 units are used, Steel-Belted Radius must request additional allocation from the billing server to allow the session to continue.

Figure 7 on page 49 illustrates this example.

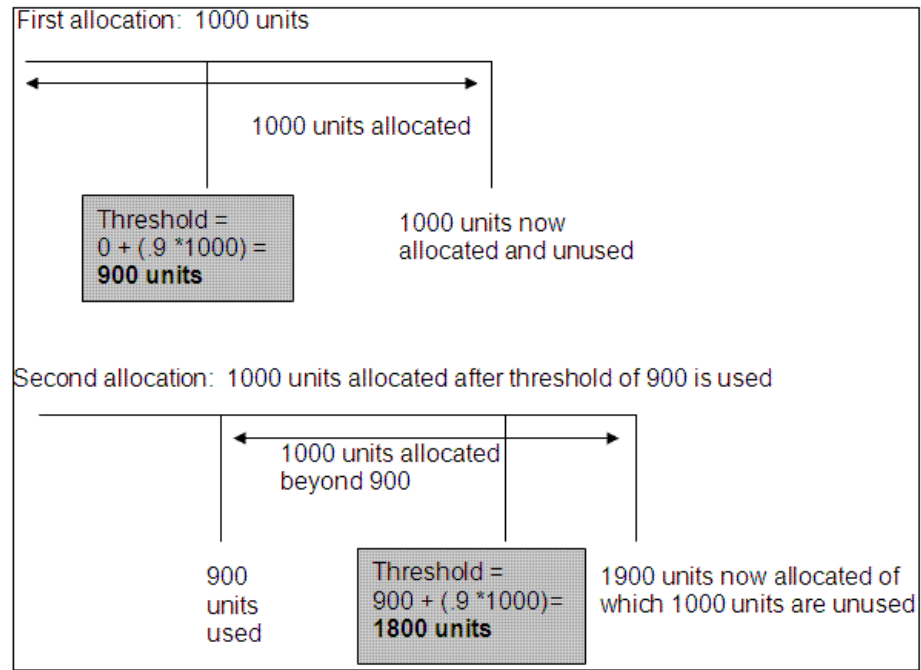


Figure 7 Threshold Calculation with Allocations of 1000 Units and Threshold Ratio Set to .9

To configure the `parlayPPSplugin.gen` file:

- 1 Open the `parlayPPSplugin.gen` file in a text editor.
- 2 In the [Settings] section, change the values for the variables listed in Table 18 on page 47 to the appropriate values.

Example

`parlayPPSplugin.gen` File

```
[Settings]
ChargingManagerLoc=corbaloc::bill_server_host/IpChargingManager
ChargingType=volume
MerchantName = "Test Merchant"
AccountId = 123987
ThresholdRatio = 0.9
VolumeUnitMultiplier = 1024
ListSessionOnHUP=True
RequestTimeoutMillliseconds=5000
```

Configuring the radius.ini File for the Prepaid Module

The `radius.ini` file must include certain lines that activate the prepaid function.

The `radius.ini` file is included with the Mobile IP module.

To configure the `radius.ini` file:

- 1 Add the following line in the `[GenericPlugins]` section to activate the prepaid server plugin.

```
parlayPPSplugin.so=parlayPPSplugin.gen
```

- 2 In the `[Configuration]` section:

Set `AttributeEdit` to 1.

Set `CheckMessageAuthenticator` to 1.

- 3 If you desire automatic timeouts, set the following variables in the `[Configuration]` section:

```
AcctAutoStopEnable=1
```

```
SessionTimeoutOnMissedAcctStopEnable=1
```

`SessionTimeoutOnMissedAcctStopGracePeriodSec` — Set to the number of extra seconds beyond `SessionTimeoutSeconds` after which a session times out. (`SessionTimeoutSeconds` is set in `prepaidAttr.att`.)

For more information about session timeouts, see [“Configuring Prepaid Timeouts” on page 54](#).

- 4 Add the word `prepaidAttr` to the `[AttributeEditing]` section.

This setting (`prepaidAttr`) matches the `InitializationString` in the `prepaidAttr.att` file and activates the `prepaidAttr.att` plugin.

Example

radius.ini file

```
[GenericPlugins]
parlayPPSplugin.so=parlayPPSplugin.gen
[Configuration]
AttributeEdit=1
CheckMessageAuthenticator=1
AcctAutoStopEnable=1
SessionTimeoutOnMissedAcctStopEnable=1
SessionTimeoutOnMissedAcctStopGracePeriodSec=30
[AttributeEditing]
prepaidAttr
```

Sending and Receiving Prepaid Attributes

Certain specific attributes and sub-types must be included with the Access-Request from the PDSN so that the prepaid request is recognized and can be properly handled by Steel-Belted Radius and the billing server. The PDSN must also recognize certain attributes that are sent by Steel-Belted Radius when an Access-Accept is sent for a prepaid request. These attributes are defined in cdma2000 Wireless IP Network

Standard, X.P0011-006-C, PrePaid Packet Data Services, Accounting Services and 3GPP2 RADIUS VSAs. v0.5. available at <http://www.3gpp2.org>.

Prepaid Attributes

Table 19 lists the Prepaid module attributes that are involved in incoming Access-Requests from the PDSN and outgoing Access-Accepts from Steel-Belted Radius to the PDSN.

Table 20 on page 52 provides additional information about the Prepaid module attributes.

Initial Access-Requests establish and authorize sessions. Online Access-Requests tell the billing server to update account balance information and ask for more allocation of volume or duration.

In Table 19, the following notation is used:

- (1) 1 indicates that the attribute must appear once.
- (2) 0 indicates that the attribute must not be present.
- (3) 0-1 indicates that the attribute may be either absent or present once.
- (4) Blank cell indicates that the presence or absence of the attribute has no effect

Table 19. Prepaid Attributes

Attribute or Subtype	Initial Access-Request	Initial Access-Accept	Online Access-Request	Online Access-Accept
3GPP2-Prepaid-Acct-Capability	1 ⁽¹⁾	1		0 ⁽²⁾
AvailableInClient	1	0	0	0
SelectedForSession	(4)	1	0	0
3GPP2-Prepaid-Acct-Quota	0	0-1	1	0-1 ⁽³⁾
QuotaIdentifier	0	0-1	1	0-1
VolumeQuota	0	0-1	0-1	0-1
VolumeQuotaOverflow	0	0-1	0-1	0-1
DurationQuota	0	0-1	0-1	0-1
VolumeThreshold	0	0-1		0-1
VolumeThresholdOverflow	0	0-1		0-1
DurationThreshold	0	0-1		0-1
UpdateReason	0	0	1	0
3GPP2-Session-Term-Capability		0-1		0
3GPP2-Session-Continue	0	0	0	0
Event-Timestamp	0-1	1	0-1	1
User-Name	1	0	1	0
Calling-Station-Id	0-1	0	0-1	0
Service-Type (Must be set to Authorize Only.)		0	1	0
3GPP2-Correlation-ID	1	0	1	0

Table 19. *Prepaid Attributes (Continued)*

Message-Authenticator	0	0	1	1
3GPP2-Service-Reference-ID	0-1		0-1	

Table 20. *Notes on Prepaid Attributes*

Attribute or Subtype	Notes
3GPP2-Prepaid-Acct-Capability	Must be present in the initial request. Sets up the Prepaid module session for the subscriber.
AvailableInClient	Indication from the PDSN of which type(s) of prepaid service are supported: <ul style="list-style-type: none"> • 1=volume • 2=duration • 3=both
SelectedForSession	Indicates which type of prepaid service has been selected. (Must be volume or duration, but not both.) The selection is determined by the ChargingType variable set in the <code>parlayPPSplugin.gen</code> file. (See “Configuring the parlayPPSplugin.gen File” on page 47.)
3GPP2-Prepaid-Acct-Quota	This attribute will be present in the Access-Request and Access-Accept if there is valid quota to be returned. When the quota is exhausted or the session is ended, this attribute is not present.
QuotaIdentifier	Uniquely identifies the assignment of quota within a session. When more quota is requested, the request must contain the same identifier.
Volume Quota or Duration Quota	Either VolumeQuota or DurationQuota must be present in a returned Access-Accept, depending on the type of prepaid service, unless the return is an empty 3GPP2-Prepaid-Acct-Quota to indicate session termination. Either VolumeQuota or DurationQuota is expected in an online Access-Request.
VolumeThreshold or Duration Threshold	Calculated amount that represents the point at which additional quota should be requested. For more information about VolumeThreshold, see “Determining the Volume or Duration Threshold” on page 48.
VolumeQuotaOverflow VolumeThresholdOverflow	VolumeQuotaOverflow will be present if VolumeQuota is present but the quota is too large to be stored in VolumeQuota alone. VolumeThresholdOverflow will be present if VolumeThreshold is present but the quota is too large to be stored in VolumeQuota.

Table 20. Notes on Prepaid Attributes (Continued)

UpdateReason	Indication of the reason quota is being requested. The following numbers are relevant to Prepaid module functionality in the Mobile IP module: <ul style="list-style-type: none"> • 3=Threshold reached • 4=Quota reached • 5=Remote Forced disconnect • 6=Client Service termination
3GPP2-Session-Term-Capability	This attribute will be present in the Initial Access-Accept if SessionTerminationCapability is set to 1, 2 or 3 in the <code>prepaidAttr.att</code> file. If SessionTerminationCapability is set to 0, then 3GPP2-Session-Term-Capability will not be present.
3GPP2-Session-Continue	Messages containing this attribute will not receive prepaid processing.
Event-Timestamp	Required for Duration-based prepaid only. If the Event-Timestamp attribute is present, ParlayPPSPlugin checks that it is current within 300 seconds. If it is not present or current, ParlayPPSPlugin silently discards the Access-Request.
User-Name	Either User-Name or Calling-Station-Id will be expected in the incoming messages to define the client's identity to the prepaid server. The choice of User-Name or Calling-Station-Id is set in the PrepaidAttr.att file. See “Configuring the prepaidAttr.att Plugin File” on page 45 for more information.
Calling Station Id	Either User-Name or Calling-Station-Id will be expected in the incoming messages to define the client's identity to the prepaid server. The choice of User-Name or Calling-Station-Id is set in the PrepaidAttr.att file. See “Configuring the prepaidAttr.att Plugin File” on page 45 for more information.
Service-Type	For online access requests, the Service-Type must be set to Authorize-Only and no password should be present.
3GPP2-Correlation-ID	Must be present in Access-Accepts. Uniquely identifies a session.
Message-Authenticator	Required for all authorize-only requests for security purposes. (All online requests are authorize-only requests.)
3GPP2-Service-Reference-ID	This attribute, if present, must be used to indicate that this is the Main Service Instance. If this attribute is present, then the Main-SC-Indicator subtype value must be set to 1.

Filtering Prepaid Attributes

If you specify a filter in the [FA-User-Auth-Requests], [SIP-User-Auth-Requests], or [Authorize-Only-Requests] sections, make sure that the filter is defined in the `filter.ini` file and that the filter allows (or does not exclude) all the attributes relevant to the Prepaid module requests. These attributes are:

For [FA-User-Auth-Requests], [SIP-User-Auth-Requests], or [Authorize-Only-Requests]:

```
3GPP2-Prepaid-Acct-Quota
3GPP2-Session-Continue
Event-Timestamp
User-Name
Calling-Station-ID
Service-Type
3GPP2-Correlation-ID
Message Authenticator
3GPP2-Service-Reference-ID
```

For [FA-User-Auth-Requests] and [SIP-User-Auth-Requests] sections, the following two attributes must be allowed in addition to the attributes listed in the previous list:

```
3GPP2-Prepaid-Acct-Capability
3GPP2-Session-Term-Capability
```

For more information about filtering for the Prepaid module, see [“Configuring Filters for Prepaid Attributes” on page 44](#).

Configuring Prepaid Timeouts

You can configure two types of timeouts for the Prepaid module:

Session Timeouts — This type of timeout applies to all Steel-Belted Radius sessions.

Request Timeouts — This type of timeout applies only to the Prepaid module Parlay interface.

Session Timeouts

Session timeouts limit the number of seconds that a session may remain active, after which an AutoStop occurs if a session Stop is not received. If you do not set a session timeout, the session ends only when a Stop is received.

To configure session timeouts:

- 1 In the `radius.ini` [Configuration] section, set the following values to 1:


```
AcctAutoStopEnable=1
SessionTimeoutOnMissedAcctStopEnable=1
```
- 2 In the [Settings] section of `prepaidAttr.att`, set the number of seconds for a session timeout using `SessionTimeoutSeconds`. For example:


```
SessionTimeoutSeconds=36000
```
- 3 In the `radius.ini` [Configuration] section, set the value for the grace period to be allotted after the official timeout. For example, if the timeout is set to 36000 seconds with `SessionTimeoutSeconds` and the grace period is 30 seconds, the session times out at 36030 seconds. For example:


```
SessionTimeoutOnMissedAcctStopGracePeriodSec=30
```

Example**prepaidAttr.att file**

```
[Settings]
SessionTimeoutSeconds=36000
```

radius.ini file

```
[Configuration]
AcctAutoStopEnable=1
SessionTimeoutOnMissedAcctStopEnable=1
SessionTimeoutOnMissedAcctStopGracePeriodSec=30
```

Request Timeouts

Request timeouts are the number of milliseconds allowed to elapse after a request from Steel-Belted Radius to the billing server for a debit or for more quota. After this amount of time, the request will timeout and an Access-Reject is returned.

To configure request timeouts:

- 1 In the [Settings] section of `ParlayPPSPlugin.gen`, use the `RequestTimeoutMillisconds` variable to set the number of milliseconds for the request timeout.

NOTE: For more information about the `ParlayPPSPlugin.gen` file, see [“Configuring the parlayPPSplugin.gen File” on page 47](#).

Example**ParlayPPSPlugin.gen file**

```
[Settings]
RequestTimeoutMillisconds=5000
```

Using New Session Hotlining and Prepaid Sessions Together

You may choose to allow Prepaid module sessions to be hotlined. For more information about new session hotlining, see [“New Session Hotlining” on page 33](#).

To configure the Mobile IP module to allow hotlining of prepaid sessions:

- 1 In the [AttributeEditing] section of `radius.ini`, add the word `prepaidAttr`. (This line must precede the line for `NSHL` in the [AttributeEditing] section.)
- 2 Make sure the word `NSHL` appears in the [AttributeEditing] section of `radius.ini`, to allow new session hotlining functionality. (This line must come after the line for `prepaidAttr` in the [AttributeEditing] section.)
- 3 In the [Settings] section of the `nshl.att` file, add the line `AllowHotliningOfPrepaidSession=1`.

If `AllowHotliningOfPrepaidSession` is set to 0, no hotlining attributes will be returned.

Example**radius.ini file**

```
[AttributeEditing]
prepaidAttr
NSHL
```

```
nshl.att file
[Settings]
AllowHotliningOfPrepaidSession=1
```

Disabling Prepaid Functionality

To disable the Prepaid module, edit the `prepaidAttr.att` file, `prepaidAcct.acc` file, and the `radius.ini` file so that Prepaid module information is not expected.

To disable prepaid functionality:

- 1 In the `prepaidAttr.att` file, set `Enable` to 0.
- 2 In the `prepaidAcct.acc` file, set `Enable` to 0.
- 3 Make the following changes to the `radius.ini` file:
 - a Remove the line `parlayPPSplugin.so=parlayPPSplugin.gen`.
 - b Remove the word `prepaidAttr` from the `[AttributeEditing]` section.

Dynamic Mobile Update

Dynamic Mobile Update (DMU) is a process for dynamically provisioning password keys on mobile nodes (MNs) and the Steel-Belted Radius database. The password keys (MN_HAAA, MN_HA, and CHAP_Key) control access of the MN to the wireless network. The assignment of these keys is either done on demand or done when the keys expire. During the key-provisioning process, the key values are protected by encryption and mutual authentication between the MN and Steel-Belted Radius. The DMU plug-in is a program that incorporates DMU functionality into Steel-Belted Radius.

DMU provides an advantage to MN manufacturers and network administrators because it allows:

- ▶ “Over the air” provisioning of password keys (rather than passwords that must be manually set) and
- ▶ Periodic reassignment of password keys (rather than permanent passwords).

NOTE: The DMU specification is an IETF (Internet Engineering Task Force) draft, *Dynamic Mobile IP Key Update for cdma2000 Networks, March 2005* (draft-carroll-dynmobileIP-cdma-05.txt). For specific information about the data exchange that occurs during the DMU process, see the DMU specification. For a copy of the DMU specification, see your Juniper Technical Services representative.

Summary of DMU Data Exchange

When Steel-Belted Radius receives an Access-Request, it determines whether the password keys have expired and new keys are needed. If new key provisioning is required, Steel-Belted Radius rejects the mobile node’s access request. The mobile node then generates new keys, encrypts them, and sends them to the Steel-Belted Radius server. Steel-Belted Radius decrypts, verifies, and records the new keys. In subsequent accesses, the mobile node uses one of the newly provisioned keys.

When new keys need to be provisioned, mutual authentication takes place during the DMU process between the mobile node and the Steel-Belted Radius server.

Configuring DMU for Steel-Belted Radius

DMU becomes an active component of Steel-Belted Radius as soon as it is configured. The following steps are necessary for configuring DMU. Each of these steps is described in the sections that follow.

To configure DMU for use with Steel-Belted Radius:

- 1 Configure the `3gpp2.ini` file.
- 2 Configure the `dmu.aut` file.
- 3 Configure the `sqlaccessor.gen` file or `sqlaccessorjdbc.gen` file.
- 4 Configure the `radius.ini` file.
- 5 Create a SQL stored procedure.
- 6 Initialize the SQL database.
- 7 Configure the Steel-Belted Radius Authentication Policies screen.

Configuring the 3GPP2.ini File for DMU

The `3gpp2.ini` file (Table 21) must contain certain settings to enable DMU functionality.

Table 21. 3GPP2.ini Settings for DMU

Setting	Description
Enable	Set to 1 to enable 3GPP2 functionality. Set to 0 to disable 3GPP2 functionality.
AddFunk3GPP2Request TypeToRequest	Set to 1 to add the Funk-3GPP2-Request-Type attribute to the request, enabling the appropriate key password obtained from the database. (See “Corresponding Request Types and Key Types” on page 69 for more information.)

Example

```
[Settings]
Enable = 1
AddFunk3GPP2RequestTypeToRequest = 1
```

Configuring the dmu.aut File

The `dmu.aut` file controls settings that affect the function of the DMU plug-in. You must configure the `dmu.aut` file to make DMU function with Steel-Belted Radius and to set up specific choices about DMU functionality.

NOTE: The `dmu.aut` file is read whenever Steel-Belted Radius restarts or receives a HUP signal.

Four sections of the `dmu.aut` file must be configured:

- ▶ [Bootstrap] section
- ▶ [Settings] section
- ▶ [DatabaseQueries] section
- ▶ [PKID_PrivateKey]section

[Bootstrap] Section of dmuaut

You enable DMU to function with an Enable field in the [Bootstrap] section (Table 22).

Table 22. [Bootstrap] Section

Setting	Description
Enable	Set to 1 to enable DMU functionality. Set to 0 to disable DMU functionality.

[Settings] Section of dmuaut

The [Settings] section (Table 23) provides information that is needed so that DMU can provision keys.

Table 23. [Settings] Section

Setting	Description
DatabaseAccessorName = <i>methodname</i>	Set to the same methodname used in the [Settings] section of <code>SQLaccessor.gen</code> . The <i>methodname</i> is the name of the SQL accessor that is used to access the SQL database.
KeysLifetimeDays	Set to the number of days until the keys expire after provisioning. If set to 0, the keys never expire.
UseMNAAuthenticator	Set to one of the following settings: <ul style="list-style-type: none"> • HRPD-Only - Indicates that if an HRPD network is detected, the mobile node will be authenticated with MNAAuthenticator in the MIP-Key-Data attribute. This is the default (recommended) setting. • Always - Indicates that the mobile node will always be authenticated with MNAAuthenticator in the MIP-Key-Data attribute. • Never - Indicates that the mobile node will never be authenticated with MNAAuthenticator in the MIP-Key-Data attribute. <p>NOTE: <i>Steel-Belted Radius uses the attribute 3GPP2-HRPD-Terminal-Auth (with a value of 1) to detect that the network is of type HRPD.</i></p>

Table 23. [Settings] Section (Continued)

Setting	Description
UseMSIDAsAuthenticator	<p>Set to one of the following settings:</p> <ul style="list-style-type: none"> • 1X-Only - Indicates that if a 1X network is detected, the mobile node will be authenticated with MSIS in the MIP-Key-Data attribute. This is the default (recommended) setting. • Never - Indicates that the mobile node will never be authenticated with MSID (device identifier assigned by the provider) in the MIP-Key-Data attribute. • Always - Indicates that the mobile node will always be authenticated with MSID in the MIP-Key-Data attribute. <p>Set to 1X-Only, Always, or Never. The default (recommended) setting is 1X-Only. Setting this field to 1X-Only indicates that if an 1X network is detected, the mobile node will be authenticated with MSIS in the MIP-Key-Data attribute.</p> <p>NOTE: <i>Steel-Belted Radius uses the absence of the attribute 3GPP2-HRPD-Terminal-Auth to detect whether the network is of type 1X. If the attribute is missing from the Access-Request or if the attribute has a value other than 1, the network is determined to be of type 1X.</i></p>
ServerPKOID	<p>Set to your organization's public key identifier. This is the 8-bit (two hexadecimal digits) number identifier of the Public Key Organization (PKO) that created the public key.</p>
PrivateKeysDirectory	<p>Set to the directory in which the private key files are stored. DMU uses a selected private key to decrypt MIP_Key_Data attribute.</p> <p>Set this value to a period to indicate that the private key file is found in the directory where Steel-Belted Radius is installed. Use a period followed by a path to set this directory to a path relative to the Steel-Belted Radius installation directory.</p> <p>Examples:</p> <pre>PrivateKeysDirectory = . PrivateKeysDirectory = ./KeyDir</pre> <p>NOTE: <i>The filenames themselves (within the directory) are specified in the [PKID_PrivateKey] section.</i></p>
PrivateKeysEncoding	<p>Set to DER or PEM to indicate the type of encoding used for the private key. The same type of encoding (DER or PEM) must be used for all the private keys in the PrivateKeysDirectory.</p>

Table 23. [Settings] Section (Continued)

Setting	Description
ConfigLog	DMU attribute. See “Dynamic Mobile Update” on page 56 for information about DMU. Valid values are: <ul style="list-style-type: none"> • ConsoleAndLog sends the log information to both the console and the log. • Console sends the log information to the console only. • Log sends the log information to the log only

Example

```
[Settings]
DatabaseAccessorName=SQLAccessor1
KeysLifetimeDays=30
UseMNAAuthenticator=HRPD-Only
UseMSISAsAuthenticator=1X-Only
ServerPKOID=0E
PrivateKeysDirectory=. \PrivKeyDir
PrivateKeysEncoding=DER
ConfigLog = ConsoleAndLog
```

[DatabaseQueries] Section of dmuaut

The [DatabaseQueries] section (Table 24) refers to the settings configured in the [Query] section of the SQLAccessor.gen file. These settings control the number to be used to select which SQL statement, configured in sqlaccessor.gen, is to be used to “select” DMU information from the database and which SQL statement is to be used to “update” DMU information in the database.

Table 24. [DatabaseQueries] Section

Setting	Description
SelectQueryNumber	Set to the same value as used in the [Query] section of the SQLAccessor.gen file for the DMUSelect statement. Example: sqlaccessor.gen file <pre>[Query] 1=DMUSelect</pre> dmuaut file <pre>[DatabaseQueries] SelectQueryNumber=1</pre>

Table 24. [DatabaseQueries] Section (Continued)

Setting	Description
UpdateQueryNumber	<p>Set to the same value as used in the [Query] section of the <code>SQLaccessor.gen</code> file for the <code>DMUUpdate</code> statement.</p> <p>Example:</p> <p>sqlaccessor.gen file</p> <pre>[Query] 2=DMUUpdate</pre> <p>dmu.aut file</p> <pre>[DatabaseQueries] UpdateQueryNumber=2</pre>

Example

```
[DataBaseQueries]
SelectQueryNumber = 1
UpdateQueryNumber = 2
```

Public Key / Private Key Pair

The PKID identifies a public key. Private keys and public keys exist in pairs. The PKID identifies a public key and therefore its corresponding private key.

The public key is loaded into the mobile device by the public key organization (PKO), usually the network operator or the device manufacturer. You save the corresponding private key in a file. The file must be placed in the directory specified with the `PrivateKeysDirectory` field in the [Settings] section of `3GPP2.ini`. The name of the file is configured in the [PKID_PrivateKey] Section of the `dmu.aut` file.

Each private key is contained in a separate file. The private keys may be in DER format or PEM format, and all keys must be of the same format. You specify the format in the [Settings] section of the `dmu.aut` file with the `PrivateKeysEncoding` field.

When a mobile node initiates a registration request, it uses its public key to encode part of the `MIP_Key_Update_Request` attribute. The Steel-Belted Radius server uses the private key to decrypt the information.

[PKID_PrivateKey] Section of dmu.aut

The [PKID_PrivateKey] section (Table 25) maps a public key identifier (PKID) to a private key.

The PKID field in the [PKID_PrivateKey] section specifies the mapping between the public key identifier and the file containing the corresponding private key identifier.

Table 25. [PKID_PrivateKey] Section

Setting	Description
PKID= <i>private_key_filename</i>	Map each public key identifier to the filename of the file that contains the corresponding private key identifier. Create a separate line for every public key identifier used in your network. NOTE: <i>The contents of the public key identifier are described in the DMU specification.</i>

PKID has the following format:

```
PKOID.PKOI.PK_Expansion.ATV.DMUV
```

where:

- ▶ **PKOID (Public Key Organization Identifier)**
Identifies an 8-bit field (two hexadecimal digits) that identifies the PKO that created the public key.
- ▶ **PKOI (Public Key Organization Index)**
Identifies an 8-bit field (two hexadecimal digits) consisting of a value defined by the public key organization. The PKOI may be used for any purpose, such as identification of a public/private key pair.
- ▶ **PK_Expansion (Public Key Expansion)**
Identifies an 8-bit field (two hexadecimal digits) used to enable possible expansion of the PKOID or PKOI fields.
- ▶ **ATV (Algorithm Type and Version)**
Identifies a 4-bit field (one hexadecimal digit) that identifies the public key algorithm. Valid values are 1 for RSA-1024, 2 for RSA-768, and 3 for RSA-2048.
- ▶ **DMUV (DMU Version)**
Identifies a 4-bit hexadecimal field (one hexadecimal digit) that specifies the DMU version. This field must be set to 0.

Example

```
dmu.aut File
[PKID_PrivateKey]
0a.00.01.01.0
1F.00.01.02.0
```

Configuring the *sqlaccessor.gen* or *sqlaccessorjdbc.gen* File for DMU

The *sqlaccessor.gen* file or *sqlaccessorjdbc.gen* file controls the interaction of the DMU plug-in with the SQL database.

[Settings] Section

The [Settings] section (Table 26) identifies the SQL accessor and the location of the SQL database.

Table 26. [Settings] Section Fields

Setting	Description
<i>Methodname</i> =SQLAccessor	The <i>methodname</i> is any name that you assign to identify the SQL accessor used to access the SQL database. The DatabaseAccessor field in the [Settings] section of the <i>dmu.aut</i> file must match this <i>methodname</i> .
Connect	Specifies the string that must be passed to the database client engine to establish a connection to the database. The format of the connect string depends on the type of database you use: Oracle: <code>Connect=dB_username/dB_password@OracleInstance</code> JDBC: <code>Connect=DSN=jdbc:provider.driver:dsn_name_here;UID=username_for_dB;PWD=password_for_dB</code> See the <i>Steel-Belted Radius Reference Guide</i> for more information.

Oracle Example

```
[Settings]
Methodname = SQL Accessor
Connect=DSN=johnsmith@secret@subscriberdb
```

JDBC Example

```
[Settings]
Methodname = SQL Accessor
Connect=DSN=mydb;UID=jsmith;PWD=secret
```

[VariableTypes] Section

The [VariableTypes] section (Table 27) identifies the variables (and their types) that are used to pass information between the DMU process and the database.

Warning: Do not make any changes to the [VariableTypes] section.

Table 27. [VariableTypes] Section

Setting	Description
NAI	Network Access Identifier. Represents the username of the mobile node user.
MSID	Mobile Station Identifier. MSID is used to authenticate mobile nodes on 1X type networks.
MNAuthenticator	Mobile Node Authenticator. This variable is used to authenticate mobile nodes on HRPD type networks.
MN_HAAA_Key	Password key provisioned by DMU to authenticate a mobile node for FA Authentication requests.
MN_HA_Key	Password key provisioned by DMU to authenticate a mobile node for HA Authentication or MN-HA Shared Key Distribution requests.
CHAP_Key	Password key provisioned by DMU to authenticate a mobile node for SIP Authentication requests.
MIPUpdateState	Indicates the state of the provisioning keys. <ul style="list-style-type: none"> • 1 indicates the provisioning keys need to be updated. • 2 indicates the provisioning keys have been updated. • 0 indicates that the provisioning keys are valid. The value can be manually set to 1 to initiate the DMU provisioning process. Otherwise, the state is set automatically.
KeyUTCExpireTime	Date and time that the provisioning keys expire, expressed in Coordinated Universal Time (UTC). Value is automatically generated by the system by adding the value of KeysLifetimeDays in the [Settings] section of the <code>dmu.aut</code> file to the date the keys were last provisioned.

Example

NOTE: The [VariableTypes] section indicates specific variable names and types (such as string or integer). Leave this section exactly as it comes in the `sqlaccessor.gen` or `sqlaccessorjdbc.gen` file and do not make any changes to this section.

```
[VariableTypes]
NAI=string
MSID=string
MNAuthenticator=string
MN_HAAA_Key=string
MN_HA_Key=string
CHAP_Key=string
MIPUpdateState=integer
KeyUTCExpireTime=integer
```

[Query] Section

The [Query] section (Table 28) indicates the SQL statement that uses “select” or “update” information in the SQL database.

Table 28. [Query] Section

Setting	Description
<i>n</i> =DMUSelect	<p>The value of <i>n</i> must match the value of SelectQueryNumber in the [DatabaseQueries] section of the <code>dmu.aut</code> file.</p> <p>Example:</p> <p>sqlaccessor.gen file</p> <pre>[Query] 1=DMUSelect</pre> <p>dmu.aut file</p> <pre>[DatabaseQueries] SelectQueryNumber=1</pre>
<i>m</i> =DMUUpdate	<p>The value of <i>m</i> must match the value of UpdateQueryNumber in the [DatabaseQueries] section of the <code>dmu.aut</code> file.</p> <p>Example:</p> <p>sqlaccessor.gen file</p> <pre>[Query] 2=DMUUpdate</pre> <p>dmu.aut file</p> <pre>[DatabaseQueries] UpdateQueryNumber=2</pre>

Example

```
[Query]
1 = DMUSelect
2 = DMUUpdate
```

[Query/DMUSelect] Section

The [Query/DMUSelect] section contains the SQL statement for the DMUSelect query used to retrieve information from the database.

The SQL statement in the [Query/DMUSelect] section has the following format:

```
SQL=SELECT msid,mnauth,haaa,ha,chap,state,expire FROM
dbname WHERE user_name=@NAI
```

where:

- ▶ *msid* = Column name of the SQL database column containing the MSID values.
- ▶ *mnauth* = Column name of the SQL database column containing the MNAuthenticator values.
- ▶ *haaa* = Column name of the SQL database column containing the MN_HAAA_Key values.
- ▶ *ha* = Column name of the SQL database column containing the MN_HA_Key values.

- ▶ *chap* = Column name of the SQL database column containing the CHAP_Key values.
- ▶ *state* = Column name of the SQL database column containing the MIPUpdateState values.
- ▶ *expire* = Column name of the SQL database column containing the KeyUTCExpireTime values.
- ▶ *databasename* = database identifier
- ▶ *user_name* = Column header of the SQL database column containing the NAI values

Example

```
[Query/DMUSelect]
SQL=SELECT MSID,MNAuth,HAAA, HA,CHAP,KeyExpire,State FROM
SQL_databasename WHERE User_Name=@NAI
```

[Results/DMUSelect] Section

The [Results/DMUSelect] section (Table 29) maps the SQL database column names to the variable names specified in the [VariableTypes] section.

Table 29. [Results/DMUSelect] Section

Setting	Description
MSID= <i>position_num</i> / <i>max_chars</i>	<i>Position_num</i> represents the position in the SQL DMUSelect statement that corresponds to the MSID variable. <i>Max_chars</i> represents the maximum number of characters in the SQL database column that contains the MSID values.
MNAuthenticator= <i>position_num</i> / <i>max_chars</i>	<i>Position_num</i> represents the position in the SQL DMUSelect statement that corresponds to the MNAuthenticator variable. <i>Max_chars</i> represents the maximum number of characters in the SQL database column that contains the MNAuthenticator values.
MN_HAAA_Key= <i>position_num</i> / <i>max_chars</i>	<i>Position_num</i> represents the position in the SQL DMUSelect statement that corresponds to the MN_HAAA_Key variable. <i>Max_chars</i> represents the maximum number of characters in the SQL database column that contains the MN_HAAA_Key values.
MN_HA_Key= <i>position_</i> <i>num</i> / <i>max_chars</i>	<i>Position_num</i> represents the position in the SQL DMUSelect statement that corresponds to the MN_HA_Key variable. <i>Max_chars</i> represents the maximum number of characters in the SQL database column that contains the MN_HA_Key values.

Table 29. [Results/DMUSelect] Section (Continued)

Setting	Description
CHAP_Key=position_num/max_chars	Position_num represents the position in the SQL DMUSelect statement that corresponds to the CHAP_Key variable. Max_chars represents the maximum number of characters in the SQL database column that contains the CHAP_Key values.
MIPUpdateState=position_num	Position_num represents the position in the SQL DMUSelect statement that corresponds to the MIPUpdateState variable.
KEYUTCExpireTime=position_num	Position_num represents the position in the SQL DMUSelect statement that corresponds to the KEYUTCExpireTime variable.

Example

```
[Results/DMUSelect]
MSID = 1/48
MNAuthenticator = 2/48
MN_HAAA_Key = 3/48
MN_HA_Key = 4/48
CHAP_Key = 5/48
MIPUpdateState = 6
KEYUTCExpireTime = 7
```

Figure 8 illustrates the relationship between the [Query/DMUSelect] section, [Results/DMUSelect] section, and the SQL database. The circled areas show that the [Results/DMUSelect] section indicates that the third position in the SQL statement provides the name of the database column (HAAA) that corresponds to the MN_HAAA_Key values.

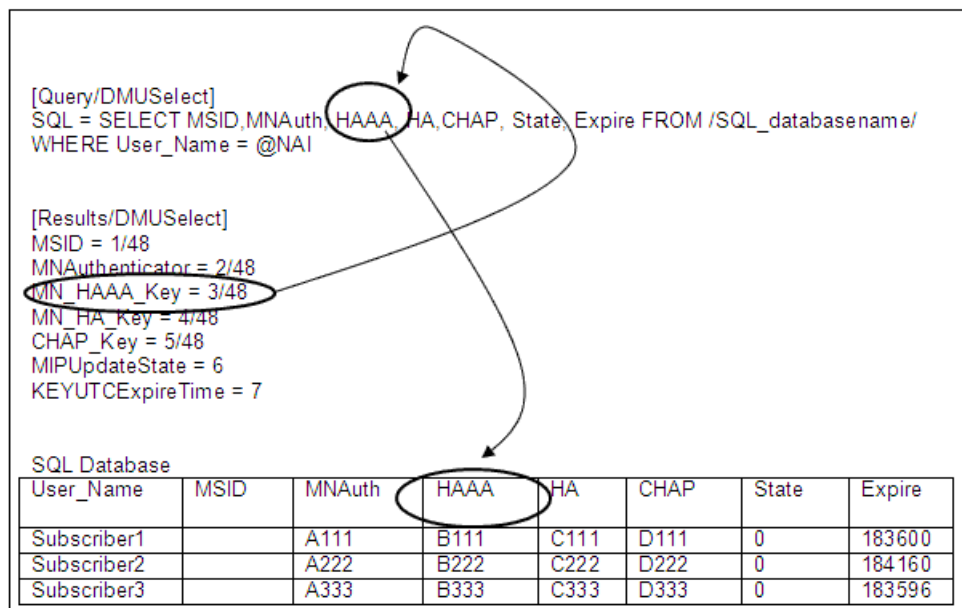


Figure 8 Mapping of Variable Names in the Results Section to the Column Names in the SQL Database

[Query/DMUUpdate] Section

The [Query/DMUUpdate] section contains the SQL statement used to write DMU information to the database. The SQL statement in this section also maps the database column names to the corresponding variables.

The SQL statement in the [Query/DMUUpdate] section has the following format:

```
SQL = UPDATE [databasename].dbo.Subscriber SET msid =@MSID,
haaa=@MN_HAAA_Key, ha=@MN_HA_Key, chap=@CHAP_Key,
state=@MIPUpdateState, expire=@KeyUTCExpireTime/n WHERE
user_name=@NAI
```

where:

- ▶ *haaa* = Column header of the SQL database column containing the MN_HAAA_Key values.
- ▶ *ha* = Column header of the SQL database column containing the MN_HA_Key values.
- ▶ *chap* = Column header of the SQL database column containing the CHAP_Key values.
- ▶ *state* = Column header of the SQL database column containing the MIPUpdateState values.
- ▶ *expire* = Column header of the SQL database column containing the KeyUTCExpireTime values.
- ▶ *user_name* = Column name of the SQL database column containing the NAI values.
- ▶ *databasename* = name of the database

NOTE: The *UPDATE* statement does not specify the *MSID* or *MNAAuthenticator* variables because these values are not written to the database by the DMU process.

Example

```
[Query/DMUUpdate]
SQL = UPDATE [SQL_databasename].dbo.Subscriber SET
HAAA=@MN_HAAA_Key, HA=@MN_HAA_Key, CHAP=@CHAP_Key,
Expire=@KeyUTCExpireTime, State=@MIPUpdateState, WHERE
User_Name=@NAI
```

NOTE: The [Query/DMUUpdate] section does not need a corresponding [Results] section because the mapping between the variable names and the column headings takes place in the SQL statement itself.

Configuring the radius.ini File for DMU

The `radius.ini` file needs to contain a section that identifies the `sqlaccessor.gen` file. The [GenericPlugins] section of the `radius.ini` file identifies the Steel-Belted Radius “plugins” (adjunct programs) and their associated generation (`.gen`) files. One of the following [GenericPlugins] section must appear in the `radius.ini` file.

Example

```
[GenericPlugins]
radiusl_accessor_ora9.so = sqlaccessor.gen

or
```

```
[GenericPlugins]
radiusl_accessor_jdbc.so = sqlaccessorjdbc.gen
```

The `radius.ini` file includes these lines (commented out). Make sure that your library names and `.gen` filenames conform to the names provided here.

Creating and Executing a Stored Procedure for DMU

You must create a stored procedure to obtain the appropriate key (MN_HAAA_Key, MN_HA_Key, or CHAP_Key) to be used to authenticate the mobile node for each request type.

For more information about the use of stored procedures, see the *Steel-Belted Radius Administration Guide* and the *Steel-Belted Radius Reference Guide*.

Corresponding Request Types and Key Types

The stored procedure must return the appropriate key for the request type. For example, if the request type is FA Authentication, the appropriate key is MN_HAAA_Key, which might correspond to the database column HAAA in your database.

The Funk-3GPP2-Request-Type attribute is assigned an integer value that indicates the request type. [Table 30](#) lists the request types, associated keys, and integer values that represent the request type

Table 30. *Corresponding Request Types, Keys, and Integer Values*

Request Type*	Integer Value of Funk-3GPP2-Request-Type**	Key Needed for Authenticating the Mobile Node
FA Authentication	2	MN_HAAA_Key
HA Key Distribution	3	N/A. A key is not updated for DMU.
MN-HA Shared Key Distribution	4	MN_HA_Key
HA Authentication	5	CHAP_Key
SIP Authentication	6	CHAP_Key

* For information on how the request type is determined, see [Figure 1 on page 10](#).

** The integer values that correspond to the request types are pre-set in the `radius.dct` file.

Stored Procedure Required Code

The stored procedure must contain lines to obtain the database value returned, based on the request type. For example, if an FA Authentication access request is detected, then the stored procedure must find the MN_HAAA_Key in the SQL database.

Syntax:

WHEN *n* THEN *colname*

where:

- ▶ *n* = integer value representing a request type. (See [Table 30](#) on page 69.)
- ▶ *colname* = column heading of the SQL database column that contains the appropriate key.

Example

```
CREATE PROC upGetUserAuthRequestPassword (
@UserName char(32),
@AuthRequestType int,
@Password char(32) OUTPUT
)
AS
BEGIN
IF EXISTS (SELECT *
FROM Subscriber
where UserName=@UserName)
    SET @Password = (
    SELECT case @AuthRequestType
        WHEN 2 THEN MN_HAAAKey -- FA Authentication
        WHEN 3 THEN '-- HA Key Distribution N/A
        WHEN 4 THEN MN_HAKEY-- MN-HA Shared Key Distribution
        WHEN 5 THEN MN_HAAAKEY-- HA Authentication
        WHEN 6 THEN CHAPKEY -- SIP Authentication
    ELSE ''
    END
    FROM Subscriber
    WHERE UserName=@UserName)
ELSE
SET @PASSWORD = ''
END
```

Stored Procedure Execution

You must place an execute statement in the `radsq1.aut` file or `radsq1jdbc.aut` file to run the stored procedure.

For more information about the execute statement, see the *Steel-Belted Radius Administration Guide*.

Example

```
SQL= EXEC upGetUserAuthRequestPassword %UserName!i,
@Funk-3GPP2-Request-Type!i, %Password!o OUTPUT;
```

Initializing the Database for DMU Processing

The SQL database must contain a column for each variable that DMU needs in its processing. Some columns may be left blank because they become populated when DMU is initiated, and information is transmitted from the mobile nodes or calculated during processing.

NOTE: *Subscriber data must be stored in SQL for DMU support.*

Table 31 shows the columns that are needed for DMU and indicates if the initial database needs to be pre-populated with values.

Table 31. *DMU Initialization Requirements for SQL Database*

Variable Represented by a Database Column	Description	Value Required (if any)
NAI	Unique username for each subscriber	Username of each subscriber.
MSID	Mobile node authenticator	For subscribers on 1X networks, populate with the MSID. For subscribers on HRPD networks, leave blank.
MNAuthenticator	Mobile node authenticator	For subscribers on HRPD networks, populate with the MNAuthenticator. For subscribers on 1X networks, leave blank.
MN_HAAA_Key	Access key provisioned for FA Authentication requests.	Leave blank.
MN_HA_Key	Access key provisioned for MN-HA Shared Key Distribution requests.	Leave blank.
CHAP_Key	Access key provisioned for SIP authentication requests.	Leave blank.
MIPUpdateState	Indicates valid keys: <ul style="list-style-type: none"> • 1 indicates keys need provisioning • 2 indicates keys are provisioned (automatically resets to 0) Value automatically resets to 1 when keys expire.	Set to 1 to initiate provisioning.
KeyUTCExpireTime	Expiration date and time expressed in universal coordinated time (UTC)	Expiration date and time expressed in universal coordinated time (UTC). Leave blank.

Configuring the Steel-Belted Radius for DMU

The Steel Belted Radius Authentication Policies screen must be set up to place DMU at the top of the list of Authentication Methods.

To configure Steel-Belted Radius to recognize the DMU plug-in:

- 1 Run Steel-Belted Radius Administrator and log into your Steel-Belted Radius server.
- 2 Click **Authentication Policies**.
- 3 Click the **Authentication Methods** tab.
- 4 Select **DMU** and click the up arrow until **DMU** appears at the top of the list (Figure 9).

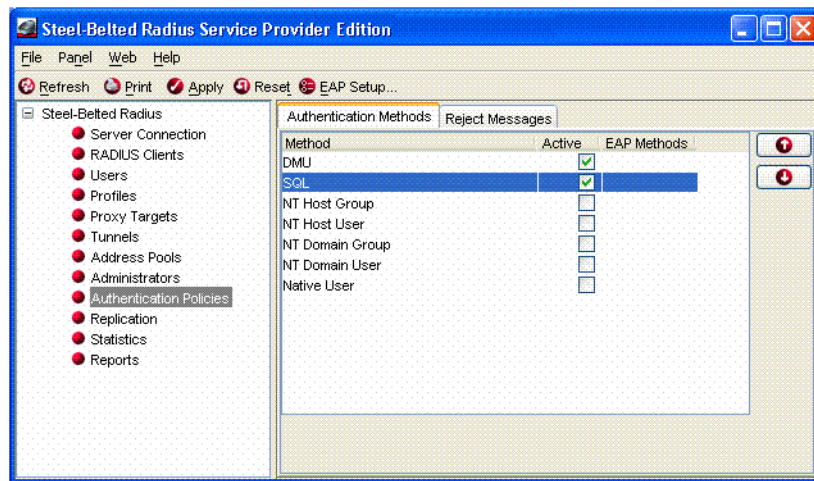


Figure 9 DMU at the Top of the Method List

LDAP Configuration Interface Extensions

With MIM enabled, you can use the LDAP Configuration Interface (LCI) to locate a session by its mobile session identifier (the value of the 3GPP2-Mobile-Correlation-Id attribute included in authentication and accounting requests).

A new node called radiusstatus=sessions_by_mobile_session has been added to the LDAP Virtual Schema. Under this node, you can specify a filter of mobile-session-id=correlation_id to locate a specific mobile session. You can also use this part of the schema to delete sessions.

Support for HRPD Access Networks

The *Interoperability Specification (IOS) for High Rate Packet Data (HRPD) Access Network Interfaces* document refers to the AAA server as the Access Network AAA (AN-AAA), and to the interface between the Packet Control Function (PCF) and the AN-AAA as the A12 interface.

NOTE: The majority of AN-AAA requirements for the A12 interface are supported by the core Steel-Belted Radius product. If Steel-Belted Radius is being deployed exclusively as an AN-AAA, you do not need to enable the Mobile IP Module.

The PCF acts as a RADIUS client device, and communicates with the AN-AAA in standard RADIUS fashion. However, support for Binary Coded Decimal (BCD) format for the Callback-Id attribute is an AN-AAA feature that requires the Mobile IP Module.

Support for this feature allows the mobile operator to provision Mobile Node Identification (MN ID) values (e.g. International Mobile Subscriber Identity - IMSI - values), which are unique to each subscriber, in human-readable form. During authentication for each session, the AN-AAA retrieves the MN ID, BCD-encodes the value, and then includes it in the Callback-Id attribute as part of the RADIUS Access-Accept response that is returned to the PCF.

After the PCF receives the RADIUS Access-Accept response containing the BCD-encoded Callback-Id, the value is interpreted by the client, and used in the A10/A11 interfaces (e.g. in communications with the PDSN infrastructure).

NOTE: If you do not know the format in which your PCF devices are expecting the Callback-Id attribute, consult your PCF vendor.

The MN IDs are typically provisioned into a scalable, highly available external database which is accessible to the AN-AAA for authentication services. For Steel-Belted Radius, both LDAP and SQL database types are supported.

NOTE: When provisioning your subscriber database, the MN ID values may consist only of decimal digits

Configuring for BCD Encoding

To configure Steel-Belted Radius for authentication against your LDAP or SQL database, refer to the *Steel-Belted Radius Administration Guide*.

NOTE: When you configure Steel-Belted Radius for LDAP or SQL authentication, you must configure the *.aut file so that the MN ID (IMSI) value, which is retrieved from the external database as part of the authentication and authorization process, is mapped to the Callback-Id, so that IMSI value will be included in the Accept-Response to the PCF.

Callback-Id, which is an IETF-defined standard RADIUS attribute, is managed in the Steel-Belted Radius dictionary radius.dct. You must add a bcd-encode tag to the default Callback-Id entry in the radius.dct dictionary (ATTRIBUTE Callback-Id 20 string r).

To modify the Callback-Id entry in radius.dct:

- 1 Stop the Steel-Belted Radius server.
- 2 Update the default Callback-Id entry in the radius.dct dictionary to include the following tag:

```
ATTRIBUTE Callback-Id 20 string r bcd-encode
```

- 3 Restart the Steel-Belted Radius server.

Chapter 4

3GPP

This chapter describes how to configure 3GPP support in the Mobile IP Module for Steel-Belted Radius.

NOTE: *Although the Mobile IP Module for Steel-Belted Radius supports both 3GPP and 3GPP2, you cannot enable 3GPP and 3GPP2 simultaneously. If 3GPP2 and 3GPP are both enabled, Steel-Belted Radius disables 3GPP and reports the configuration error in the radius log file.*

Overview

The Mobile IP Module for Steel-Belted Radius extends RADIUS functionality to 3GPP implementation on the scale required by Internet Service Providers and carriers. 3GPP support facilitates the management of mobile sessions and their associated resources through communication with a Gateway GPRS Support Node (GGSN). 3GPP support in the Mobile IP Module is based on the specifications given in the *Interworking between the Public Land Mobile Network (PLMN) supporting Packet Based Services and Packet Data Networks (PDN)* documentation (TS 29.061), which is available at www.3gpp.org.

Within a 3GPP network, the AAA server interfaces with only one type of RADIUS client device, called a Gateway GPRS Support Node. The 3GPP specification allows for a GGSN to generate multiple Accounting Start/Stop pairs for a given session. When 3GPP is enabled, the Steel-Belted Radius server with MIM expects the GGSN to send the 3GPP-Session-Stop-Indicator attribute as needed in Accounting Stop requests.

- ▶ When a GGSN sends an Accounting Stop request that includes a 3GPP-Session-Stop-Indicator attribute (regardless of its value), the GGSN tells the AAA server that the session should be deleted.
- ▶ When a GGSN sends an Accounting Stop request that does not include a 3GPP-Session-Stop-Indicator attribute in the request, the Steel-Belted Radius server with MIM marks the session as dormant and does not free any of its allocated resources (e.g., IP address).

This type of Accounting Stop request is typically followed by an Accounting Start that marks the session as being active again.

A single 3GPP user may have multiple sessions (multiple PDP contexts) on a single GGSN. Steel-Belted Radius with the Mobile IP Module supports the ability to

differentiate between these sessions by using a concatenation of the value of the User-Name attribute with the one-character NSAPI (network service access point identifier) value.

Product Registration

The Mobile IP Module is fully integrated into Steel-Belted Radius. If you have purchased MIM, a single license key activates both Steel-Belted Radius and MIM. When you install Steel-Belted Radius, the system requests that you type in the license key. After the license key is registered, you should edit the configuration settings, as explained in the sections in this chapter.

MIM Module Configuration

NOTE: When you purchase the Mobile IP Module, you receive a feature upgrade license key. You must add this key to Steel-Belted Radius and restart the server for the Mobile IP Module to load. Please see the *Steel-Belted Radius/Service Providers Edition Administration Guide* for instructions on how to add the new license key.

The `3gpp.ini` file, which contains the 3GPP settings for the Mobile IP Module, consists of the following fixed sections:

Table 32. *3gpp.ini* Sections

[Settings]	Enables or disables 3GPP processing.
[Attributes]	Identifies dictionary attributes necessary for request processing.

[Settings] Section

The [Settings] section (Table 33) contains the master switch that enables the 3GPP feature set:

Table 33. [Settings] Section

Setting	Description
Enable	Set this field to 1 to enable support for this module. Specifying 0 disables this module.

Example

```
[Settings]
Enable = 1
```

[Attributes] Section

The [Attributes] section (Table 34) specifies the two attribute names that are used by Steel-Belted Radius when the MIM processes special accounting requests for session and resource management.

Table 34. [Attributes] Section

Setting	Description
Session-Stop-Indicator	Specifies the name of the attribute that SBR + MIM expects in Accounting Stop requests when the mobile session is to be deleted (closed).
NSAPI	Specifies the name of the attribute that SBR + MIM uses in conjunction with the username to facilitate the identification of different mobile sessions for the same user on a single GGSN. In the absence of an NSAPI, Steel-Belted Radius uses the contents of the User-Name attribute as the matching key.

Example

```
[Attributes]
Session-Stop-Indicator = 3GPP-Session-Stop-Indicator
NSAPI = 3GPP-NSAPI
```

The radius.dct file shipped with Steel-Belted Radius has been configured with the attributes necessary for supporting Mobile IP services in compliance with the 3GPP standards.

NOTE: You should use the default settings in this section unless your GGSN vendor indicates that different settings must be used.

LDAP Configuration Interface Enhancements

With MIM enabled, you can use the LDAP Configuration Interface (LCI) to locate a session by its mobile session identifier (the concatenation of the values of the User-Name and 3GPP-NSAPI attributes included in authentication and accounting requests).

A new node called radiusstatus=sessions_by_mobile_session has been added to the LDAP Virtual Schema. Under this node, you can specify a filter of mobile-session-id=session_id to locate a specific mobile session. You can also use this part of the schema to delete sessions.

Numerics

- 3GPP 75
- 3GPP2 1, 57
- 3gpp2.ini file 20, 27, 31, 32, 34, 43, 57
- 3GPP2-Correlation-Id attribute 5, 9, 11, 19
- 3GPP2-Home-Agent-Address attribute 8, 11, 12
- 3GPP2-Key-ID attribute 11
- 3GPP2-MN-HA-SPI attribute 8
- 3GPP2-Pre-Shared-Secret attribute 11
- 3GPP2-Pre-Shared-Secret-Request attribute 8, 11, 16
- 3GPP2-Security-Level attribute 16
- 3GPP2-Session-Continue attribute 3, 4
- 3GPP2-S-Request attribute 8, 17

A

- Accept-Requests field 23, 25, 26
- access type, determining 10
- Access-Accept response 6, 7, 11
- accounting starts 3
- accounting stops 3
- AddFunk3GPP2RequestTypeToRequest 21
- address management, IP 2
- address pools for dynamic HA 12
- Attributes section 21, 27
- ATV 62
- Auth field 24
- Auth-Prefix field 24

B

- BCD encoding 73
- Bootstrap section 46, 58

C

- capability attribute 37
- cdma2000 1, 41, 50
- CHAP_Key variable 64
- CHAP-Challenge attribute 11
- CHAP-Password 17, 18, 19

- CHAP-Password attribute 11
- Check-NAS-IP-Address 17, 24
- check-pool 15
- Check-Source-Address 24
- ConfigLog 60
- Connect field 63

D

- DatabaseAccessorName 58
- DatabaseQueries section 60
- databases 71
- disabling new session hotlining 36
- DMU. See dynamic mobile update
- dmu.aut file 57
- DMUQuery 65
- DMUSelect 65
- DMUV 62
- dotted-quad representation 17
- duration threshold 48
- duration-based prepaid services 42
- dynamic HA assignment 23
- dynamic home agent 12, 28, 31
- dynamic mobile update 56

E

- enabling
 - 3GPP 1
 - 3GPP2 1, 21, 30, 43
 - DMU 57, 58
 - dynamic HA 23
 - FA user requests 11
 - HA key distribution requests 23
 - HA user requests 19, 25
 - inter-PDSN handoff 22, 32
 - MN-HA shared key distribution requests 25
 - new session hotlining 34
 - SIP user requests 6, 26

F

- FA requests 69

FA user authentication requests 7, 11
FA-User-Auth-Requests section 20, 23
FA-User-Auth-Requests/name] section 27
Filter field 22, 23, 24, 25, 26
filtering
 hotlining capabilities 37
 prepaid attributes 44, 53
filters 6, 16, 19, 37, 44
foreign agent 3
Framed-IP-Address attribute 14
FramedIPAddressHint field 14

G

Gateway GPRS Support Node 75

H

HA key distribution requests 7, 17, 69
HA requests 69
HA user authentication requests 7, 19
HA. See home agent
HA-Address 25
HA-Address-Mismatch 12, 23
HA-Address-Round-Robin-Group 23
HA-Key-Distribution-Requests section 20, 23
handoff, inter-pdsn 31
HAs section 21, 27
HA-User-Auth-Requests section 20, 25
hexadecimal notation 17
home address assignment process 14
home agent 3
home agent assignment process 12
home agent assignment summary 13, 31
hotlining
 new session 33
 prepaid sessions 35, 55
HRPD networks 72

I

initial Access-Requests 51
inter-PDSN handoff 12, 14, 15, 28, 31
IP address management 2
IPsec security 16

K

KeysLifetimeDays 58
KeyUTCExpireTime variable 64

L

LDAP configuration 28, 72
LDAP configuration interface for 3GPP 77

M

methodname 63
MIPSessions ForDevice
 MustHaveSameHomeAndHAAAddress 22
MIPUpdateState variable 64
MN_HA_Key variable 64
MN_HAAA_Key variable 64
MNAAuthenticator variable 64
MN-HA shared key distribution request 7
MN-HA shared key distribution requests 18, 69
MN-HA-Shared-Key-Requests section 20, 25
MN-HA-SPI attribute 18
mobile IP 2
mobile IP request types 6
MSID variable 64

N

NAI variable 64
NAS-IP-Address attribute 11, 17
new session hotlining 33
NSAPI 77
nshl.att file 34

O

online Access-Requests 51
other requests 5, 7
Other-Requests section 21, 26

P

Parlay specifications 41
parlayPPSPlugin.gen file 42
PDSN defined 2
PK_Expansion 62

- PKOI 62
- PKOID 62
- pools
 - Home Address 14
 - Home Agent 29
 - IP address 19
 - round robin 12, 29
- prepaid data services 41
- prepaid module
 - 3gpp2.ini file 43
 - attributes 50
 - components 42
 - configuring 43
 - disabling 56
 - hotlining sessions 55
 - overview 41
 - parlayPPSplugin.gen file 47
 - prepaidAcct.acc plugin 45
 - prepaidAttr.att plugin 45
 - radius.ini file 50
 - timeouts 54
- prepaidAcct.acc file 42
- prepaidAttr.att file 42
- private key 61
- PrivateKeysDirectory 59
- PrivateKeysEncoding 59
- proxy realm based configuration 16
- public key 61

Q

- Query section 65
- Query/DMUSelect section 65
- Query/DMUUpdate section 68

R

- radius.ini file 50, 68
- radsq1.aut file 39
- radsq1jdbc.aut file 39
- request timeouts 54
- requests, determining type 10
- requests, mobile IP 6
- Results/DMUSelect section 66
- roaming 3
- round robin 30
- round robin group 23

S

- ServerPKOID 59
- session timeouts 54
- Session-Stop-Indicator 77
- Settings section 21, 46, 58, 63
- simple IP request types 5
- SIP requests 69
- SIP-User-Auth-Requests section 21, 26
- S-Key 7, 11
- SQL statements 65
- sqlaccessor.gen file 63
- sqlaccessorjdbc.gen file 63
- S-Seconds field 21
- stored procedures 69

T

- threshold 48
- timeouts 54

U

- UniqueDevice Identifier 22
- UseMNAAuthenticator 58
- UseMSIDAsAuthenticator 59
- User-Name attribute 11, 17, 18, 19
- User-Password 17, 18, 19
- Use-S-Request-As-Marker 17, 24

V

- VariableTypes section 64
- volume threshold 48
- volume-based prepaid services 42

W

- weighted pools of addresses 12

