



## **Cisco Prime Access Registrar 8.0 User Guide**

Published: January 25, 2018

Last Modified: January 25, 2018

**Cisco Systems, Inc.**  
[www.cisco.com](http://www.cisco.com)

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

*Cisco Prime Access Registrar 8.0 User Guide*  
© 2018 Cisco Systems, Inc. All rights reserved.



---

**CHAPTER 1****Overview 1-1**

- Prime Access Registrar Hierarchy 1-2
  - UserLists and Groups 1-2
  - Profiles 1-3
  - Scripts 1-3
  - Services 1-3
- Session Management Using Resource Managers 1-4
- RADIUS PROTOCOL 1-5
  - Types of RADIUS Messages 1-5
    - Packet Contents 1-5
    - The Attribute Dictionary 1-6
- Related Documentation 7
- Obtain Documentation and Submit a Service Request 1-7
- 1-7

---

**CHAPTER 2****Using the Graphical User Interface 2-1**

- Launching the GUI 2-1
  - Disabling HTTP 2-2
  - Disabling HTTPS 2-2
  - Login Page 2-3
    - Logging In 2-3
    - Logging Out 2-4
- Common Methodologies 2-4
  - Filtering Records 2-4
  - Editing Records 2-5
  - Deleting Records 2-5
  - Setting Record Limits per Page 2-6
  - Performing Common Navigations 2-6
  - Relocating Records 2-7
- Dashboard 2-8
  - Sessions 2-8
- Configuring Cisco Prime Access Registrar 2-9
  - RADIUS 2-10
    - Setting Up or Changing the Radius Properties 2-11

- Profiles **2-11**
  - Adding Profile Details **2-12**
- UserGroups **2-12**
  - Adding UserGroup Details **2-14**
- UserList **2-14**
  - Adding UserList Details **2-15**
- Users **2-16**
  - Adding User Details **2-17**
- Scripts **2-18**
  - Adding Script Details **2-23**
- Policies **2-23**
  - Adding Policy Details **2-23**
- GroupServers **2-24**
  - Adding Group Server Details **2-25**
- Services **2-25**
  - Simple Services **2-26**
  - ServiceWithRS **2-34**
  - PEAP Service **2-38**
  - EAP Service **2-41**
  - Diameter Service **2-52**
- CommandSets **2-57**
  - Adding a Command Set **2-57**
- DeviceAccessRules **2-58**
  - Adding a Device Access Rule **2-58**
- FastRules **2-59**
  - Adding a Fast Rule **2-59**
- Replication **2-60**
  - Adding Replication Details **2-61**
  - Adding the Replication Member Details **2-61**
- RADIUSDictionary **2-61**
  - Adding RADIUS Dictionary Details **2-62**
- VendorDictionary **2-63**
  - Adding Vendor Dictionary Details **2-64**
- Vendor Attributes **2-64**
  - Adding Vendor Attributes **2-65**
- Vendors **2-65**
  - Adding Vendor Details **2-66**
- Translations **2-67**
  - Adding Translation Details **2-68**
- TranslationGroups **2-68**

Adding Translation Group Details	2-69
Diameter	2-69
General	2-70
Session Management	2-72
Applications	2-73
Commands	2-74
DiameterAttributes	2-76
Advanced	2-77
Default	2-77
BackingStore/ServerParam	2-83
RemoteSessionServer	2-88
SNMP and Server Monitor	2-90
DDNS	2-94
Encrypted IMSI Private Keys	2-95
ODBC DataSources	2-96
Log	2-97
Ports	2-100
Interfaces	2-101
Attribute Groups	2-101
Rules	2-102
Setting Rules	2-103
SessionManagers	2-104
Adding Session Manager Details	2-107
ResourceManager	2-108
Adding Resource Manager Details	2-116
Network Resources	2-116
Clients	2-116
Adding Client Details	2-122
Remote Servers	2-123
LDAP	2-123
LDAP Accounting	2-127
ODBC/OCI	2-130
ODBC/OCI-Accounting	2-132
Diameter	2-135
REST	2-139
Others	2-141
Administration	2-146
Administrators	2-147
Adding Administrator Details	2-147
Statistics	2-148

- Resetting Server Statistics 2-151
- DiameterStatistics 2-152
- TACACSStatistics 2-161
- Back Up and Restore 2-162
- LicenseUpload 2-163
- Read-Only GUI 2-163

**CHAPTER 3**

**RADIUS Accounting 3-1**

- Understanding RADIUS Accounting 3-1
- Setting Up Accounting 3-2
  - Accounting Log File Rollover 3-2
    - FilenamePrefix 3-3
    - MaxFileSize 3-4
    - MaxFileAge 3-4
    - RolloverSchedule 3-5
    - UseLocalTimeZone 3-5
    - FileType 3-5
    - EnableRolloverIntelligence 3-5
    - AttributesToBeLogged 3-6
- Oracle Accounting 3-6
  - Configuring Oracle Accounting 3-7
    - ODBC-Accounting Service 3-7
    - ODBC RemoteServers 3-7
    - Configuration Examples 3-9
  - Packet Buffering 3-10
    - When Using Packet Buffering 3-11
    - With Packet Buffering Disabled 3-11
  - Dynamic SQL Feature 3-11
- LDAP Accounting 3-12
  - Configuring LDAP Accounting 3-12
    - LDAP-Accounting Service 3-12
    - LDAP RemoteServers 3-13
    - Configuration Examples 3-15
    - Configuring the LDAP Service for Accounting 3-16
    - Configuring an LDAP-Accounting RemoteServer 3-17
    - Setting LDAP-Accounting As Accounting Service 3-19
- MySQL Support 3-20
  - Configuring MySQL 3-20
  - Example Configuration 3-21

Proxying Accounting Records	3-21
Configuring the Local Cisco Prime Access Registrar Server	3-22
Configuring the Local Accounting Service	3-22
Configuring the Remote Accounting Service	3-22
Configuring the Group Accounting Service	3-23
Configuring the RemoteServer Object	3-23

**CHAPTER 4****Diameter 4-1**

Diameter with EAP Support	4-2
Advertising Application Support	4-2
Diameter EAP Conversation Flow	4-2
Diameter Server Startup Log	4-3
Diameter Stack Level Messages	4-4
Capabilities Exchange Message	4-5
Watchdog Message	4-6
Disconnect Message	4-6
Configuring Authentication and Authorization for Diameter	4-6
Configuring Local Authentication and Authorization	4-6
Configuring a Local Service and UserList	4-7
Configuring External Authentication Service	4-8
Configuring the Diameter Application in Prime Access Registrar	4-8
Configuring the Transport Management Properties	4-9
Registering Applications IDs	4-10
Configuring the Diameter Peers	4-11
Configure the Diameter Service	4-12
Group-Based Load Balancing in Diameter Proxy Server Configuration	4-15
Writing Diameter Application in Prime Access Registrar	4-17
Configuring rex script/service for Diameter	4-17
Scripting in Diameter	4-18
Diameter Environment Variables	4-18
Sample rex script/service	4-19
Traces/Logs	4-20
Translation Framework for Diameter	4-21
TLS Support for Diameter	4-22
Managing Diameter Sessions	4-24
Blacklisting Support for Diameter Remote Server	4-24
SCTP Multihoming Support for Diameter Client and Remote Server	4-24

**CHAPTER 5**

**Extensible Authentication Protocols 5-1**

- EAP-AKA 5-2
  - Configuring EAP-AKA 5-2
  - Testing EAP-AKA with radclient 5-6
- EAP-AKA-Prime (EAP-AKA') 5-6
  - Configuring EAP-AKA' 5-7
  - Testing EAP-AKA' with radclient 5-8
- EAP-FAST 5-8
  - Configuring EAP-FAST 5-9
  - EAP-FAST Keystores 5-13
  - Testing EAP-FAST with radclient 5-13
    - PAC Provisioning 5-14
    - Authentication 5-15
  - Parameters Used for Certificate-Based Authentication 5-15
    - radclient Command Reference 5-16
  - PAC—Credential Export Utility 5-18
    - PAC Export 5-18
    - PAC Display 5-19
    - Syntax Summary 5-19
- EAP-GTC 5-19
  - Configuring EAP-GTC 5-19
  - Testing EAP-GTC with radclient 5-20
- EAP-LEAP 5-21
  - Configuring EAP-LEAP 5-21
- EAP-MD5 5-22
  - Configuring EAP-MD5 5-22
- EAP-Negotiate 5-23
  - Configuring EAP-Negotiate 5-23
  - Negotiating PEAP Tunnel Services 5-24
  - Testing EAP-Negotiate with radclient 5-24
- EAP-MSChapV2 5-24
  - Configuring EAP-MSChapV2 5-24
  - Testing EAP-MSChapV2 with radclient 5-25
- EAP-SIM 5-26
  - Configuring EAP-SIM 5-26
  - Quintets to Triplets Conversion 5-30
- EAP-Transport Level Security (TLS) 5-31
  - Configuring EAP-TLS 5-31

Configuring EAP-TLS with OCSP Support	5-34
Testing EAP-TLS with RSA or ECC Certificate using radclient	5-35
Testing EAP-TLS with Client Certificates	5-35
EAP-TTLS	5-35
Configuring EAP-TTLS	5-36
Creating an EAP-TTLS Service	5-36
Configuring an EAP-TTLS Authentication Service	5-39
Testing EAP-TTLS with radclient	5-42
Testing EAP-TTLS Using Legacy Methods	5-43
Testing EAP-TTLS Using EAP Methods	5-43
rehash-ca-certs Utility	5-44
radclient Command Reference	5-44
eap-trace	5-45
tunnel	5-45
Protected EAP	5-46
PEAP Version 0	5-46
Configuring PEAP Version 0	5-46
Testing PEAP Version 0 with radclient	5-50
Testing PEAP Version 0 with Client Certificates	5-50
PEAP Version 1	5-51
Configuring PEAP Version 1	5-51
Testing PEAP Version 1 with radclient	5-53
Testing PEAP Version 1 with Client Certificates	5-54
How to Configure Oracle, Mysql Accounting with the Buffering Option Enabled	5-54
To Select the SQL Statement in Run Time Accounting	5-54
Query	5-54
Insert	5-55
Update	5-55
Delete	5-55
Configuring Oracle, Mysql Accounting	5-56
How Suffix and Prefix Rules Work with Prime Access Registrar	5-57
Configuring Prefix and Suffix Policies	5-57
CRL Support for Cisco Prime Access Registrar	5-58
Configuring Certificate Validation Using CRL	5-59
Using Intermediate Certificates in Prime Access Registrar	5-59
Rolling Encryption Support for Pseudonym Generation in EAP-SIM, EAP-AKA, and EAP-AKA' Services	5-61
Support for Decrypting Encrypted-IMSI for EAP-SIM, EAP-AKA, and EAP-AKA' Services	5-64
Extended-EAP Support in Prime Access Registrar	5-66

**CHAPTER 6**

**Using Replication 6-1**

- Replication Overview 6-1
- How Replication Works 6-2
  - Replication Data Flow 6-3
    - Master Server 6-3
    - Slave Server 6-3
  - Security 6-4
  - Replication Archive 6-4
  - Ensuring Data Integrity 6-4
    - Transaction Data Verification 6-4
    - Transaction Order 6-5
    - Automatic Resynchronization 6-5
  - Full Resynchronization 6-5
  - Understanding Hot-Configuration 6-6
  - Replication's Impact on Request Processing 6-6
- Replication Configuration Settings 6-6
  - RepType 6-7
  - RepTransactionSyncInterval 6-7
    - Master 6-7
    - Slave 6-7
  - RepTransactionArchiveLimit 6-8
  - RepIPAddress 6-8
  - RepPort 6-8
  - RepSecret 6-8
  - ReplsMaster 6-9
  - RepMasterIPAddress 6-9
  - RepMasterPort 6-9
  - Rep Members Subdirectory 6-9
  - Rep Members/Slave1 6-9
    - Name 6-9
    - IPAddress 6-9
    - Port 6-10
- Setting Up Replication 6-10
  - Configuring The Master 6-10
  - Configuring The Member 6-11
  - Verifying the Configuration 6-12
- Replication Example 6-13
  - Adding a User 6-13
    - Master Server's Log 6-13

Member Server's Log	6-13
Verifying Replication	6-14
Master Server's Log	6-14
Member Server's Log	6-14
Using aregcmd -pf Option	6-14
Master Server's Log	6-15
Member Server's Log	6-15
An Automatic Resynchronization Example	6-16
Master Server's Log	6-16
Member Server's Log	6-17
Full Resynchronization	6-17
Replication Setup with More Than One Slave	6-19

**CHAPTER 7****Using Identity Caching 7-1**

Overview	7-1
Identity Caching Features	7-2
Configuring Cisco Prime Access Registrar for Identity Caching	7-3
Starting Identity Caching	7-6
XML Interface	7-8

**CHAPTER 8****Using Prepaid Billing 8-1**

Overview	8-2
IS835C Prepaid Billing	8-2
Configuring IS835C Prepaid Billing	8-3
Setting Up a Prepaid Billing RemoteServer	8-3
Setting Up an IS835C Prepaid Service	8-4
Setting Up Local Authentication	8-5
Setting Up an Authentication Group Service	8-5
CRB Prepaid Billing	8-7
Configuring CRB Prepaid Billing	8-8
Setting Up a Prepaid Billing RemoteServer	8-8
Setting Up a CRB Prepaid Service	8-9
Setting Up a Local Accounting Service	8-11
Setting Up a Local Authentication Service	8-12
Setting Up a Prepaid Accounting Group Service	8-13
Setting Up an Authentication Group Service	8-14
Configuring CRB Prepaid Billing for SSG	8-15
Generic Call Flow	8-18

- Access-Request (Authentication) 8-19
- Access-Accept (Authentication) 8-20
- Access-Request (Authorization) 8-20
- Access-Accept (Authorization) 8-21
- Accounting-Start 8-22
- Data Flow 8-22
- Access-Request (Quota Depleted) 8-22
- Accept-Accept (Quota Depleted) 8-23
- Accounting Stop (Session End) 8-23
- Accounting Response (Final Status) 8-23
- Vendor-Specific Attributes 8-25
- Implementing the Prepaid Billing API 8-27

**CHAPTER 9**

**Using Cisco Prime Access Registrar Server Features 9-1**

- Incoming Traffic Throttling 9-2
  - MaximumIncomingRequestRate 9-2
  - MaximumOutstandingRequests 9-2
- Backing Store Parsing Tool 9-3
- Configurable Worker Threads Enhancement 9-4
- Session-Key Lookup 9-5
- Query-Notify 9-6
  - Call Flow 9-7
  - Configuration Examples 9-8
  - Memory and Performance Impact 9-9
- Support for Windows Provisioning Service 9-9
  - Call Flow 9-10
  - Example Configuration 9-10
    - Environment Variables 9-11
    - Master URL Fragments 9-11
  - Unsupported Features 9-12
    - Account Expiration and Renewal 9-12
    - Password Changing and Force Update 9-13
- Command Completion 9-13
- Service Grouping Feature 9-14
  - Configuration Example - AccountingGroupService 9-14
    - Summary of Events 9-17
  - Configuration Example 2 - AuthenticationGroupService 9-17
    - Summary of Events 9-20

SHA-1 Support for LDAP-Based Authentication	9-21
Remote LDAP Server Password Encryption	9-21
Dynamic Password Encryption	9-22
Logs	9-23
Dynamic Attributes	9-23
Object Properties with Dynamic Support	9-23
Dynamic Attribute Format	9-25
Tunneling Support Feature	9-25
Configuration	9-26
Example	9-26
Notes	9-27
Validation	9-27
xDSL VPI/VCI Support for Cisco 6400	9-27
Using User-Name/User-Password for Each Cisco 6400 Device	9-27
Format of the New User-Name Attribute	9-28
Apply Profile in Cisco Prime Access Registrar Database to Directory Users	9-28
User-Profile	9-28
User-Group	9-29
Example User-Profile and User-Group Attributes in Directory User Record	9-30
Directory Multi-Value Attributes Support	9-30
MultiLink-PPP (ML-PPP)	9-30
Dynamic Updates Feature	9-31
NAS Monitor	9-33
Automatic Information Collection (arbug)	9-33
Running arbug	9-33
Files Generated	9-34
Simultaneous Terminals for Remote Demonstration	9-34
Support for RADIUS Check Item Attributes	9-34
Configuring Check Items	9-35
User-Specific Attributes	9-36
Packet of Disconnect	9-36
Configuring Packet of Disconnect	9-37
Configuring the Client Object	9-37
Configuring a Resource Manager for POD	9-38
Proxying POD Requests from External Servers	9-38
CLI Options for POD	9-39
query-sessions	9-39
release-sessions	9-39

- Configuring Change of Authorization Requests 9-40
  - Configuring the Client Object 9-40
- Dynamic DNS 9-41
  - Configuring Dynamic DNS 9-42
  - Testing Dynamic DNS with radclient 9-44
- Dynamic Service Authorization Feature 9-45
  - Configuring Dynamic Service Authorization Feature 9-45
    - Setting Up the Environment Variable 9-46
- Remote Session Management 9-48
- Wx Interface Support for SubscriberDB Lookup 9-49
  - Configuration Examples 9-49
- Smart Grid Solution Management 9-51
- Lawful Interception (LI) Support in Prime Access Registrar 9-51
  - Configuring Lawful Intercept 9-56
- TACACS+ Support for AAA 9-57

**CHAPTER 10**

- Directing RADIUS Requests 10-1**
  - Configuring Policies and Rules 10-1
    - Configuring Policies 10-1
    - Configuring Rules 10-2
    - Wildcard Support 10-2
    - Script and Attribute Requirements 10-3
    - Validation 10-4
    - Known Anomalies 10-4
  - Routing Requests 10-4
    - Routing Requests Based on Realm 10-4
    - Routing Requests Based on DNIS 10-5
    - Routing Requests Based on CLID 10-6
    - Routing Requests Based on NASIP 10-7
    - Routing Requests Based on User-Name Prefix 10-8
    - Attribute Translation 10-9
      - Translations 10-9
      - TranslationGroups 10-9
      - Parsing Translation Groups 10-10
    - Time of Day Access Restrictions 10-11
      - Setting Time Ranges in ExecTimeRule 10-12
      - ExecTimeRule Example Configuration 10-12
      - Reducing Overhead Using Policies to Group Rules 10-13

Standard Scripts Used with Rules	10-15
ExecRealmRule	10-15
ExecDNISRule	10-16
ExecCLIDRule	10-16
ExecNASIPRule	10-17
ExecPrefixRule	10-17
ExecSuffixRule	10-18
Configuring Suffix and Prefix Policies	10-19
ExecTimeRule	10-20
ParseTranslationGroupsByRealm	10-20
ParseTranslationGroupsByDNIS	10-20
ParseTranslationGroupsByCLID	10-21
ParseTranslationGroupsByDNIS	10-21

**CHAPTER 11****Using FastRules to Process Packet Flow 11-1**

Configuring FastRules	11-2
-----------------------	------

**CHAPTER 12****Using LDAP 12-1**

Configuring LDAP	12-1
Configuring the LDAP Service	12-2
MultipleServersPolicy	12-2
RemoteServers	12-3
Configuring an LDAP RemoteServer	12-3
DNS Look Up and LDAP Rebind Interval	12-6
LDAPToRadiusMappings	12-7
LDAPToEnvironmentMappings	12-7
LDAPToCheckItemMappings	12-7
Setting LDAP As Authentication and Authorization Service	12-7
Saving Your Configuration	12-7
CHAP Interoperability with LDAP	12-8
Allowing Special Characters in LDAP Usernames	12-8
Dynamic LDAP Search Base	12-8
Analyzing LDAP Trace Logs	12-9
Successful Bind Message	12-9
Bind Failure Messages	12-9
Login Failure Messages	12-10
Bind-Based Authentication for LDAP	12-11

**CHAPTER 13**

**Using Open Database Connectivity 13-1**

- Oracle Software Requirements 13-2
- Configuring ODBC/OCI 13-2
  - Configuring an ODBC/OCI Service 13-6
  - Configuring an ODBC/OCI RemoteServer 13-7
    - OCI Connection Timeout and Disconnection 13-9
    - ODBC Data Source 13-10
    - SQL Definitions 13-10
    - SQL Syntax Restrictions 13-11
    - Specifying More Than One Search Key 13-12
    - ODBCToRadiusMappings/OCIToRadiusMappings 13-12
    - ODBCToEnvironmentMappings/OCIToEnvironmentMappings 13-12
    - ODBCToCheckItemMappings/OCIToCheckItemMappings 13-13**
  - Configuring an ODBC DataSource 13-13
  - Setting ODBC/OCI As Authentication and Authorization Service 13-13
  - Setting ODBC/OCI As Accounting Service 13-14
  - Saving Your Configuration 13-14
  - Oracle Stored Procedures 13-14
- MySQL Support 13-16
  - MySQL Driver 13-16
  - Configuring a MySQL Datasource 13-17
  - Example Configuration 13-18

**CHAPTER 14**

**SIGTRAN-M3UA 14-1**

- Prerequisites to SIGTRAN-M3UA 14-2
- Configuring EAP-AKA/EAP-SIM with SIGTRAN-M3UA 14-4
  - ANSI Support for SIGTRAN 14-7
- Configuring M3UA Service 14-13
  - Configuring M3UA Service with Map Restore Data Authorization 14-14
    - Map Restore Data Authorization Flow 14-14
    - CS Insert Subscriber Data Structure 14-15
    - CLI Configuration for Map-Restore-Data 14-16
- Blacklisting Support for SIGTRAN-M3UA Remote Server 14-21
- Support for SCTP Multihoming in SIGTRAN-M3UA 14-21
- Tuning Global SIGTRAN Parameters 14-22
- SIGTRAN-M3UA Logs 14-24

**CHAPTER 15****Using SNMP 15-1**

## Overview 15-1

## Supported MIBs 15-1

RADIUS-AUTH-CLIENT-MIB 15-2

RADIUS-AUTH-SERVER-MIB 15-2

RADIUS-ACC-CLIENT-MIB 15-2

RADIUS-ACC-SERVER-MIB 15-2

CISCO-DIAMETER-BASE-PROTOCOL-MIB 15-2

Diameter SNMP and Statistics Support 15-3

TACACS+ SNMP and Statistics Support 15-3

## SNMP Traps 15-3

## Supported Traps 15-4

carServerStart 15-5

carServerStop 15-5

carInputQueueFull 15-5

carInputQueueNotVeryFull 15-5

carDialInputQueueFull 15-5

carDialInputQueueNotFull 15-6

carOtherAuthServerNotResponding 15-6

carOtherAuthServerResponding 15-6

carOtherAccServerNotResponding 15-7

carOtherAccServerResponding 15-7

carAccountingLoggingFailure 15-7

carLicenseUsage 15-8

carSigtranLicenseUsage 15-8

carDiameterPeerDown 15-8

carDiameterPeerUp 15-8

carTPSCapacityFull 15-8

carTPSCapacityNotFull 15-9

carSigtranTPSCapacityFull 15-9

carSigtranTPSCapacityNotFull 15-9

carSessionCapacityFull 15-9

carSessionCapacityNotFull 15-10

carSigtranSessionCapacityFull 15-10

carSigtranSessionCapacityNotFull 15-10

carLicenseUsageReset 15-11

carSigtranLicenseUsageReset 15-11

carReplicationSyncFailure 15-11

TLSCClientConnectionUpTrap 15-11





# Overview

---

The chapter provides an overview of the RADIUS server, including connection steps, RADIUS message types, and using Cisco Prime Access Registrar (Prime Access Registrar) as a proxy server.

Prime Access Registrar is a 3GPP-compliant, 64-bit carrier-class RADIUS (Remote Authentication Dial-In User Service)/Diameter server that enables multiple dial-in Network Access Server (NAS) devices to share a common authentication, authorization, and accounting database.

Prime Access Registrar handles the following tasks:

- Authentication—determines the identity of users and whether they can be allowed to access the network
- Authorization—determines the level of network services available to authenticated users after they are connected
- Accounting—keeps track of each user’s network activity
- Session and resource management—tracks user sessions and allocates dynamic resources

Using a RADIUS server allows you to better manage the access to your network, as it allows you to store all security information in a single, centralized database instead of distributing the information around the network in many different devices. You can make changes to that single database instead of making changes to every network access server in your network.

Prime Access Registrar also allows you to manage the complex interconnections of the new network elements in order to:

- adequately manage the traffic
- perform appropriate load balancing for desired load distribution
- allow binding of different protocol interfaces corresponding to a subscriber/network element

Service providers transform their 3G and 4G wireless networks with complex services, tiered charging, converged billing, and more by introducing increasing numbers and types of Diameter-based network elements. LTE and IMS networks are the most likely to implement these new network elements—including Policy and Charging Rules Functions (PCRF), Home Subscriber Servers (HSS), Mobility Management Entities (MME), Online Charging Systems (OCS), and others. As a result, as the traffic levels grow, these wireless networks are becoming more difficult to manage and scale without the Prime Access Registrar infrastructure.

Prime Access Registrar allows GUI-based, CLI-based, and REST API-based configurations. For more details, see [Chapter 2, “Using the Graphical User Interface”](#), “Using the aregcmd Commands” chapter of the *Cisco Prime Access Registrar 8.0 Administrator Guide*, and “REST API Framework” chapter of the *Cisco Prime Access Registrar 8.0 Reference Guide*.

# Prime Access Registrar Hierarchy

Prime Access Registrar's operation and configuration is based on a set of *objects*. These objects are arranged in a hierarchical structure much like the Windows 95 Registry or the UNIX directory structure. Prime Access Registrar's objects can themselves contain subobjects, just as directories can contain subdirectories. These objects include the following:

- Radius—the root of the configuration hierarchy
- UserLists—contains individual UserLists which in turn contain users
- UserGroups—contains individual UserGroups
- Users—contains individual authentication or authorization details of a user
- Clients—contains individual Clients
- Vendors—contains individual Vendors
- Scripts—contains individual Scripts
- Policies—contains a set of rules applied to an Access-Request
- Services—contains individual Services
- CommandSets—contains commands and the action to perform during Terminal Access Controller Access-Control System Plus (TACACS+) command authorization
- DeviceAccessRules—contains conditions or expressions and the applicable command sets for TACACS+ command authorization
- FastRules—provides a mechanism to easily choose the right authentication, authorization, accounting, and query service(s), drop, reject, or break flows, choose session manager or other rules required for processing a packet
- SessionManagers—contains individual Session Managers
- ResourceManagers—contains individual Resource Managers
- Profiles—contains individual Profiles
- RemoteServers—contains individual RemoteServers
- Advanced—contains Ports, Interfaces, Reply Messages, and the Attribute dictionary

This section contains the following topics:

- [UserLists and Groups](#)
- [Profiles](#)
- [Scripts](#)
- [Services](#)
- [Session Management Using Resource Managers](#)

## UserLists and Groups

Prime Access Registrar lets you organize your user community through the configuration objects **UserLists**, **users**, and **UserGroups**.

- Use **UserLists** to group users by organization, such as Company A and Company B. Each list contains the actual names of the users.

- Use **Users** to store information about particular users, such as name, password, group membership, base profile, and so on.
- Use **UserGroups** to group users by function, such as PPP, Telnet, or multiprotocol users. Groups allow you to maintain common authentication and authorization requirements in one place, and have them referenced by many users.

For more information about **UserLists** and **UserGroups**, see the “Configuring and Monitoring the RADIUS Server” chapter of the *Cisco Prime Access Registrar 8.0 Administrator Guide*.

## Profiles

Prime Access Registrar uses **Profiles** that allow you to group RADIUS attributes to be included in an Access-Accept packet. These attributes include values that are appropriate for a particular user class, such as PPP or Telnet user. The user’s base profile defines the user’s attributes, which are then added to the response as part of the authorization process.

Although you can use Group or Profile objects in a similar manner, choosing whether to use one rather than the other depends on your site. If you require some choice in determining how to authorize or authenticate a user session, then creating specific profiles, and specifying a group that uses a script to choose among the profiles is more flexible. In such a situation, you might create a default group and then write a script that selects the appropriate profile based on the specific request. The benefit to this technique is each user can have a single entry, and use the appropriate profile depending on the way they log in.

For more information about **Profiles**, see the “Configuring and Monitoring the RADIUS Server” chapter of the *Cisco Prime Access Registrar 8.0 Administrator Guide*.

## Scripts

Prime Access Registrar allows you to create scripts you can execute at various points within the processing hierarchy.

- Incoming scripts—enable you to read and set the attributes of the request packet, and set or change the Environment dictionary variables. You can use the environment variables to control subsequent processing, such as specifying the use of a particular authentication service.
- Outgoing scripts—enable you to modify attributes returned in the response packet.

For more information about **Scripts**, see the “Configuring and Monitoring the RADIUS Server” chapter of the *Cisco Prime Access Registrar 8.0 Administrator Guide*.

## Services

Prime Access Registrar uses *Services* to let you determine how authentication, authorization, and/or accounting are performed.

For example, to use Services for authentication:

- When you want the authentication to be performed by the Prime Access Registrar RADIUS server, you can specify the **local** service. In this case you must specify a specific **UserList**.
- When you want the authentication performed by another server, which might run an independent application on the same or different host than your RADIUS server, you can specify either a **radius**, **ldap**, or **tacacs-udp** service. In this case, you must list these servers by name.

When you have specified more than one authentication service, Prime Access Registrar determines which one to use for a particular Access-Request by checking the following:

- When an incoming script has set the Environment dictionary variable **Authentication-Service** with the name of a Service, Prime Access Registrar uses that service.
- Otherwise, Prime Access Registrar uses the default authentication service. The default authentication service is a property of the **RADIUS** object.

Prime Access Registrar chooses the authentication service based on the variable **Authentication-Service**, or the default. The properties of that Service, specify many of the details of that authentication service, such as, the specific user list to use or the specific application (possibly remote) to use in the authentication process.

For more information about Services, see the “Configuring and Monitoring the RADIUS Server” chapter of the *Cisco Prime Access Registrar 8.0 Administrator Guide*.

## Session Management Using Resource Managers

Prime Access Registrar lets you track user sessions, and/or allocate dynamic resources to users for the lifetime of their session. You can define one or more Session Managers, and have each one manage the sessions for a particular group or company.

Session Managers use Resource Managers, which in turn manage resources of a particular type as described below.

- IP-Dynamic—manages a pool of IP addresses and allows you to dynamically allocate IP addresses from that pool
- IP-Per-NAS-Port—allows you to associate ports to specific IP addresses, and thus ensure each NAS port always gets the same IP address
- IPX-Dynamic—manages a pool of IPX network addresses
- Subnet-Dynamic—manages a pool of subnet addresses
- Group-Session-Limit—manages concurrent sessions for a group of users; that is, it keeps track of how many sessions are active and denies new sessions after the configured limit has been reached
- User-Session-Limit—manages per-user concurrent sessions; that is, it keeps track of how many sessions each user has and denies the user a new session after the configured limit has been reached
- Home-Agent—manages a pool of on-demand IP addresses
- USR-VPN—manages Virtual Private Networks (VPNs) that use USR NAS Clients
- Home-Agent-IPv6—manages a pool of on-demand IPv6 addresses
- Remote-IP-Dynamic—manages a pool of IP addresses that allows you to dynamically allocate IP addresses from a pool of addresses. It internally works with a remote ODBC database.
- Remote-User-Session-Limit—manages per-user concurrent sessions; that is, it keeps track of how many sessions each user has and denies the user a new session after the configured limit has been reached. It internally works with a remote ODBC database.
- Remote-Group-Session-Limit—manages concurrent sessions for a group of users; that is, it keeps track of how many sessions are active and denies new sessions after the configured limit has been reached. It internally works with a remote ODBC database.
- Session Cache—allows you to define the RADIUS attributes to store in cache.
- Dynamic-DNS—manages the DNS server.

- Remote-Session-Cache—allows you to define the RADIUS attributes to store in cache. It should be used with session manager of type 'remote'.
- 3GPP—allows you to define the attribute for 3GPP authorization.

For more information about Session Managers, see the “Configuring and Monitoring the RADIUS Server” chapter of the *Cisco Prime Access Registrar 8.0 Administrator Guide*.

If necessary, you can create a complex relationship between the Session Managers and the Resource Managers.

When you need to share a resource among Session Managers, you can create multiple Session Managers that refer to the same Resource Manager. For example, if one pool of IP addresses is shared by two departments, but each department has a separate policy about how many users can be logged in concurrently, you might create two Session Managers and three Resource Managers. One dynamic IP Resource Manager that is referenced by both Session Managers, and two concurrent session Resource Managers, one for each Session Manager.

In addition, Prime Access Registrar lets you pose queries about sessions. For example, you can query Prime Access Registrar about which session (and thus which NAS-Identifier, NAS-Port and/or User-Name) owns a particular resource, as well as query Prime Access Registrar about how many resources are allocated or how many sessions are active.

## RADIUS PROTOCOL

- [Types of RADIUS Messages](#)

### Types of RADIUS Messages

The client/server packet exchange consists primarily of the following types of RADIUS messages:

- Access-Request—sent by the client (NAS) requesting access
- Access-Reject—sent by the RADIUS server rejecting access
- Access-Accept—sent by the RADIUS server allowing access
- Access-Challenge—sent by the RADIUS server requesting more information in order to allow access. The NAS, after communicating with the user, responds with another Access-Request.

When you use RADIUS accounting, the client and server can also exchange the following two types of messages:

- Accounting-Request—sent by the client (NAS) requesting accounting
- Accounting-Response—sent by the RADIUS server acknowledging accounting

This section contains the following topics:

- [Packet Contents](#)
- [The Attribute Dictionary](#)

### Packet Contents

The information in each RADIUS message is encapsulated in a UDP (User Datagram Protocol) data packet. A packet is a block of data in a standard format for transmission. It is accompanied by other information, such as the origin and destination of the data.

Table 1-1 lists a description of the five fields in each message packet.

**Table 1-1 RADIUS Packet Fields**

Fields	Description
Code	Indicates message type: Access-Request, Access-Accept, Access-Reject, Access-Challenge, Accounting-Request, or Accounting-Response.
Identifier	Contains a value that is copied into the server's response so the client can correctly associate its requests and the server's responses when multiple users are being authenticated simultaneously.
Length	Provides a simple error-checking device. The server silently drops a packet if it is shorter than the value specified in the length field, and ignores the octets beyond the value of the length field.
Authenticator	Contains a value for a Request Authenticator or a Response Authenticator. The Request Authenticator is included in a client's Access-Request. The value is unpredictable and unique, and is added to the client/server shared secret so the combination can be run through a one-way algorithm. The NAS then uses the result in conjunction with the shared secret to encrypt the user's password.
Attribute(s)	Depends on the type of message being sent. The number of attribute/value pairs included in the packet's attribute field is variable, including those required or optional for the type of service requested.

## The Attribute Dictionary

The Attribute dictionary contains a list of preconfigured authentication, authorization, and accounting attributes that can be part of a client's or user's configuration. The dictionary entries translate an attribute into a value Prime Access Registrar uses to parse incoming requests and generate responses. Attributes have a human-readable name and an enumerated equivalent from 1-255.

Sixty three standard attributes exist, which are defined in RFC 2138 and 2139. There also are additional vendor-specific attributes that depend on the particular NAS you are using.

Some sample attributes include:

- User-Name—the name of the user
- User-Password—the user's password
- NAS-IP-Address—the IP address of the NAS
- NAS-Port—the NAS port the user is dialed in to
- Framed Protocol—such as SLIP or PPP
- Framed-IP-Address—the IP address the client uses for the session
- Filter-ID—vendor-specific; identifies a set of filters configured in the NAS
- Callback-Number—the actual callback number.

## Related Documentation

For a complete list of Cisco Prime Access Registrar documentation, see the [Cisco Prime Access Registrar 8.0 Documentation Overview](#).

**Note**

---

We sometimes update the documentation after original publication. Therefore, you should also review the documentation on Cisco.com for any updates.

---

## Obtain Documentation and Submit a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see [What's New in Cisco Product Documentation](#).

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the [What's New in Cisco Product Documentation RSS feed](#). The RSS feeds are a free service.





## Using the Graphical User Interface

---

Cisco Prime Access Registrar (Prime Access Registrar) is a Remote Authentication Dial-In User Service (RADIUS) / Diameter server that enables multiple dial-in Network Access Server (NAS) devices to share a common authentication, authorization, and accounting database.

This chapter describes how to use the standalone graphical user interface (GUI) of Prime Access Registrar to:

- Configure Cisco Prime Access Registrar
- Manage Network Resources managed by Prime Access Registrar
- Administer Prime Access Registrar related activities

The following topics help you to work with and understand the Prime Access Registrar GUI:

- [Launching the GUI](#)
- [Common Methodologies](#)
- [Dashboard](#)
- [Configuring Cisco Prime Access Registrar](#)
- [Network Resources](#)
- [Administration](#)
- [Read-Only GUI](#)

### Launching the GUI

Prime Access Registrar requires you to use Microsoft Internet Explorer 8.0 SP1 (Windows 2000 and Windows XP). You start the GUI by pointing your browser to the Prime Access Registrar server and port 8080, as in the following:

```
http://ar_server_name:8080
```



#### Note

---

You can also use Mozilla Firefox 16.0 and Google Chrome 23.0 browsers to launch the Prime Access Registrar GUI. It can be launched using IPv6 address also.

---

To start a secure socket layer (SSL) connection, use **https** to connect to the Prime Access Registrar server and port 8443, as in the following:

```
https://ar_server_name:8443
```

By default, both HTTP and HTTPS are enabled. The following sections describe how to disable HTTP and HTTPS:

- [Disabling HTTP](#)
- [Disabling HTTPS](#)


**Note**

For proper function of Prime Access Registrar GUI, the DNS name resolution for the server's hostname should be defined precisely.

## Disabling HTTP

To disable HTTP access, you must edit the **server.xml** file in the **/cisco-ar/apache-tomcat-8.5.16/conf** directory. You must have root privileges to edit this file.

Use a text editor such as **vi** to open the **server.xml** file, and comment out lines 96-99. Use the **<!--** character sequence to begin a comment. Use the **-->** character sequence to end a comment.

The following are lines 93-99 of the **server.xml** file:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
  <!-- CHANGE MADE: Note: to disable HTTP, comment out this Connector -->
<Connector port="8080" maxHttpHeaderSize="8192"
    maxThreads="150 minSpare/Threads="25" maxSpareThreads="75"
    enableLookups="false" redirectPort="8443" acceptCount="100"
    connectionTimeout="20000" disableUploadTimeout="true" />
```

The following example shows these lines with beginning and ending comment sequences to disable HTTP:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
  <!-- CHANGE MADE: Note: to disable HTTP, comment out this Connector -->
<!--
<Connector className="org.apache.catalina.connector.http.HttpConnector"
    port="8080" minProcessors="5" maxProcessors="75"
    enableLookups="true" redirectPort="8443"
    acceptCount="10" debug="0" connectionTimeout="60000" />
-->
```

After you modify the **server.xml** file, you must restart the Prime Access Registrar server for the changes to take effect. Use the following command line to restart the server:

```
/opt/CSCOAr/bin/arserver restart
```

## Disabling HTTPS

To disable HTTPS access, you must edit the **server.xml** file in the **/cisco-ar/apache-tomcat-8.5.16/conf** directory. You must have root privileges to edit this file.

Use a text editor such as **vi** to open the **server.xml** file, and comment out lines 116-121. Use the **<!--** character sequence to begin a comment. Use the **-->** character sequence to end a comment.

The following are lines 111-121 of the **server.xml** file:

```
<!-- Define an SSL HTTP/1.1 Connector on port 8443 -->
  <!-- CHANGE MADE: enabled HTTPS.
```

```

    Note: to disable HTTPS, comment out this Connector -->
<Connector port="8443" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="true" disableUploadTimeout="true"
    acceptCount="100" scheme="https" secure="true"
    clientAuth="false"
    keystoreFile="/cisco-ar/certs/tomcat/server-cert.p12"
    keystorePass="cisco" keystoreType="PKCS12" sslProtocol="TLS" />
</Connector>

```

The following example shows these lines with beginning and ending comment sequences to disable HTTPS.

```

<!-- Define an SSL HTTP/1.1 Connector on port 8443 -->
<!-- CHANGE MADE: enabled HTTPS.
    Note: to disable HTTPS, comment out this Connector -->
<!--
<Connector className="org.apache.catalina.connector.http.HttpConnector"
    port="8443" minProcessors="5" maxProcessors="75"
    enableLookups="true"
    acceptCount="10" debug="0" scheme="https" secure="true">
    <Factory className="org.apache.catalina.net.SSLServerSocketFactory"
        keystoreFile="/cisco-ar/certs/tomcat/server-cert.p12"
        keystorePass="cisco" keystoreType="PKCS12"
        clientAuth="false" protocol="TLS" />
</Connector>
-->

```

After you modify the **server.xml** file, you must restart the Prime Access Registrar server for the changes to take effect. Use the following command line to restart the server:

```
/opt/CSCOar/bin/arserver restart
```

## Login Page

The login page has fields for a username and password. This page displays when you first attempt to log into the system, if a session times out, or after you log out of the system.

## Logging In

Users who are configured as Administrators can log into the Prime Access Registrar server.



### Note

While logging in, do not enable the save password option in the browser.

## Logging in

To log into the Prime Access Registrar GUI:

- 
- Step 1** Enter the relevant url in the browser. The Prime Access Registrar Login page is displayed.
  - Step 2** Enter the credentials in the provided fields.

**Step 3** Click **Login**. The Prime Access Registrar main page is displayed.

---

**Note**

After installation of Prime Access Registrar server, when you log into the application for the first time, the application redirects to the change password page.

---

**Refreshing the pages using the GUI**

To stop the server (when it is running), and then immediately start the server, click the **Reload** link.

**Restarting the GUI**

To restart the Prime Access Registrar server, click the **Restart** link.

**Note**

If aregcmd interface is active, then it needs to be closed for restarting the Prime Access Registrar server.

---

## Logging Out

To log out of the Prime Access Registrar GUI, click **Logout** in the upper right portion of the Prime Access Registrar GUI window.

## Common Methodologies

This section explains the operations that are common across the GUI interface of Prime Access Registrar. The functions explained in this section are referred throughout to this help system.

This section describes the following:

- [Filtering Records](#)
- [Deleting Records](#)
- [Setting Record Limits per Page](#)
- [Performing Common Navigations](#)
- [Relocating Records](#)

## Filtering Records

To filter a record:

---

**Step 1** Navigate to the required page. For example, choose **Configuration > Profiles**. The Profile page is displayed.

**Step 2** Enter the known details of the record in the **Filter** text box.

- Step 3** Click **Go**. The matching records are displayed in the search criteria below.
- Step 4** Click **Clear Filter** to clear the performed filter.
- 

You can also perform the following:

- [Deleting Records](#)
- [Editing Records](#)
- [Setting Record Limits per Page](#)
- [Performing Common Navigations](#)
- [Relocating Records](#)

## Editing Records

To edit the required records:

---

- Step 1** Navigate to the required page.
- Step 2** Search for a record using the filter option, if required.
- Step 3** Choose the required record that you want to edit.
- Step 4** Click **Edit**. The selected record details are displayed in the appropriate page.
- Step 5** Make the necessary changes.
- Step 6** Click **Submit** or **Update** to save the details. The page is displayed with the updated details and a message is prompted. Otherwise click **Cancel** to return to the page without saving the details.
- 

You can also perform the following:

- [Filtering Records](#)
- [Deleting Records](#)
- [Setting Record Limits per Page](#)
- [Performing Common Navigations](#)
- [Relocating Records](#)

## Deleting Records

To delete a record:

---

- Step 1** Navigate to the required page. For example, choose **Configuration > Profiles**. The Profile page is displayed.
- Step 2** Search for a record using the filter option, if required.

- Step 3** Check the check box against the record that you want to delete.
- Step 4** Click **Delete**. A message is displayed on successful deletion of the record.
- 

You can also perform the following:

- [Filtering Records](#)
- [Editing Records](#)
- [Setting Record Limits per Page](#)
- [Performing Common Navigations](#)
- [Relocating Records](#)

## Setting Record Limits per Page

To set the numbers of records to be displayed per page, select the record limit from the list available and click the **Go** button. The available denominations are **10, 25, 50, 100, and All**.

You can also perform the following:

- [Filtering Records](#)
- [Editing Records](#)
- [Deleting Records](#)
- [Performing Common Navigations](#)
- [Relocating Records](#)

## Performing Common Navigations

On existence of more records that cannot be accommodated in a page, the records are displayed in multiple pages. [Table 2-1](#) describes the icons used for page navigation.

**Table 2-1** *Page Navigation Icons*

Icons	Description
	To view the next page
	To return back to previous page
	To view the last page
	To return to the first page

You can also perform the following:

- [Filtering Records](#)
- [Editing Records](#)
- [Deleting Records](#)
- [Setting Record Limits per Page](#)
- [Relocating Records](#)

## Relocating Records

Table 2-2 describes the icons used for relocating records.

**Table 2-2** *Icons for Relocating Records*

Icons	Description
	To move a record from the Available List to the Selected List
	To move a record from the Selected List to the Available List
	To move all the records from the Available List to the Selected List
	To move all the records from the Selected List to the Available List
	To move the selected record one step above
	To move the selected record one step below
	To move the selected record to the first position
	To move the selected record to the last position

You can also perform the following:

- [Filtering Records](#)
- [Editing Records](#)
- [Deleting Records](#)
- [Setting Record Limits per Page](#)
- [Performing Common Navigations](#)

# Dashboard

The dashboard of the Prime Access Registrar GUI shows you the overview on the status on the server and user session details. It consists of the three tabs: **Server Status**, **User Sessions**, and **System Information**.

The **Server Status** provides the following details:

- AAA Server status— includes the AAA Process, Process ID, and Status.
- Health status of the AAA Server— the status of the AAA Server with respect to the performance condition is displayed.

The **User Sessions** consists of two graphs.

- Number of Sessions versus Duration in Days
- Number of Sessions versus Duration in Weeks

The Number of Sessions vs Duration in Weeks report provides the session details with respect to the number of weeks for which it is queried. The Number of Sessions vs Duration in Days report provides the session details with respect to the number of days for which it is queried. The Time(mins) vs Username report provides the accumulated time with respect to the selected username. This report can also be viewed in the form of chart and grid. Click the relevant icons below the graph to view the details in the respective formats.

The **System Information** section consists of two graphs:

- Disk Availability for Prime Access Registrar Directory
- CPU Utilization

The Disk Availability for Prime Access Registrar Directory report provides the details of the available disk space and used disk space in the Prime Access Registrar directory. When you hover the mouse on the pie chart, the details of the disk space are displayed. The CPU Utilization report provides the utilization of the CPU for a specific time. The CPU usage is represented in kilobits per seconds.

## Sessions

The Sessions feature of the dashboard helps you in viewing the records based on session id. [Table 2-3](#) lists and describes the various session views in the page.

**Table 2-3**      *Different Session Views*

Fields	Description
Release	Click to release the selected session details.
Release All	Click to release all the records from the list.
Send CoA	Click to send the CoA packet to the client device.
SendPoD	Click to send the disconnect packet to the NAS to clear sessions and an Accounting-Stop notification to the client listed in the session record.
Query All Sessions	Click to query all the sessions in the server.

To view sessions details:

- 
- Step 1** Choose **Dashboard > Sessions**. The Sessions page appears.
- Step 2** Choose the required session id to view **Release, Release All, Send CoA, Send PoD, and Query All Session** details. The session details are displayed as described in the above table.



---

**Note** You can locate the session id using the filter option. See [Filtering Records](#) for more details.

---

## Configuring Cisco Prime Access Registrar

Prime Access Registrar's operation and configuration are based on a set of objects. On configuring the Prime Access Registrar major components, the server objects can be created. These objects include the following:

- [RADIUS](#)— the root of the configuration hierarchy
- [Profiles](#)—contains individual Profiles
- [UserGroups](#)—contains individual UserGroups
- [UserList](#)—contains individual UserLists which in turn contain users
- [Users](#)—contains individual authentication or authorization details of a user
- [Scripts](#)—contains individual Scripts
- [Policies](#)—contains a set of rules applied to an Access-Request
- [GroupServers](#)—contains Diameter remote server groups to enable group-based load balancing among Diameter peers
- [Services](#)—contains individual Services
- [CommandSets](#)—contains commands and the action to perform during Terminal Access Controller Access-Control System Plus (TACACS+) command authorization
- [DeviceAccessRules](#)—contains conditions or expressions and the applicable command sets for TACACS+ command authorization
- [FastRules](#)—provides a mechanism to easily choose the right authentication, authorization, accounting, and query service(s), drop, reject, or break flows, choose session manager or other rules required for processing a packet
- [Replication](#)—maintains identical configurations on multiple machines simultaneously
- [RADIUSDictionary](#)—passes information between a script and the RADIUS server, or between scripts running on a single packet
- [VendorDictionary](#)—allows to maintain the attributes of the vendor with respect to vendor id, vendor type and the attributes required to support the major NAS
- [Vendor Attributes](#)—communicates prepaid user balance information from the Prime Access Registrar server to the AAA client, and actual usage, either interim or total, between the NAS and the Prime Access Registrar server

- **Vendors**—contains individual Vendors
- **Translations**—adds new attributes to a packet or change an existing attribute from one value to another.
- **TranslationGroups**—add translation groups for different user groups
- **SessionManagers**—contains individual Session Managers
- **ResourceManager**—contains individual Resource Managers
- **Remote Servers**—contains individual Remote Servers
- **Diameter**—contains Session Management, Applications, Commands, Diameter Attributes
- **Rules**—allows to set rules for service selection

## RADIUS

The **Radius** object is the root of the hierarchy. For each installation of the Cisco Prime Access Registrar server, there is one instance of the **Radius** object. You reach all other objects in the hierarchy from the **Radius**.

Table 2-4 lists and describes the fields in the Radius Properties page.



### Note

Fields which are represented with the term “required” in the windows of the Prime Access Registrar GUI, denote mandatory input.

**Table 2-4**      **Radius Properties**

Fields	Description
Name	Required; must be unique in the list of servers in the cluster.
Version	Required; the currently installed version of Prime Access Registrar.
Description	Optional; description of the server.
DefaultSessionManager	Cisco Prime Access Registrar uses this property if none of the incoming scripts sets the environment dictionary variable <b>Session-Manager</b> .  This field is mandatory if you are upgrading to a later version of Prime Access Registrar.
IncomingScript	Optional; if there is a script, it is the first script Cisco Prime Access Registrar runs when it receives a request from any client and/or for any service.
OutgoingScript	Optional; if there is a script, it is the last script Cisco Prime Access Registrar runs before it sends a response to any client.
DefaultAuthenticationService	Optional; Cisco Prime Access Registrar uses this property when none of the incoming scripts sets the environment dictionary variable <b>Authentication-Service</b> .
DefaultAuthorizationService	Optional; Cisco Prime Access Registrar uses this property when none of the incoming scripts sets the environment dictionary variable <b>Authorization-Service</b> .

**Table 2-4** *Radius Properties (continued)*

Fields	Description
DefaultAccountingService	Optional; Cisco Prime Access Registrar uses this property when none of the incoming scripts sets the environment dictionary variable <b>Accounting-Service</b> .
DefaultSessionService	Cisco Prime Access Registrar uses this property when none of the incoming scripts sets the environment dictionary variable <b>Session-Service</b> .  This field is mandatory if you are upgrading to a later version of Prime Access Registrar.

## Setting Up or Changing the Radius Properties

To set or change the Radius properties:

- 
- Step 1** Choose **Configuration > Radius**. The Radius Properties page appears.
  - Step 2** Specify the relevant details.
  - Step 3** Click **Save** to save the changes made to the Radius properties page.
- On successful setting up of the RADIUS, a message is displayed.
- 

## Profiles

You use Profiles to group RADIUS attributes that belong together, such as attributes that are appropriate for a particular class of PPP or Telnet user. You can reference profiles by name from either the **UserGroup** or the **User** properties. Thus, if the specifications of a particular profile change, you can make the change in a single place and have it propagated throughout your user community.

Although you can use UserGroups or Profiles in a similar manner, choosing whether to use one rather than the other depends on your site. When you require some choice in determining how to authorize or authenticate a user session, then creating specific profiles, and creating a group that uses a script to choose among them is more flexible.

In such a situation, you might create a default group, and then write a script that selects the appropriate profile based on the specific request. The benefit to this technique is each user can have a single entry, and use the appropriate profile depending on the way they log in.

[Table 2-5](#) lists and describes the fields in the Add Profiles page.

**Table 2-5** *Profile Properties*

Fields	Description
Name	Required; must be unique in the Profiles list.
Description	Optional; description of the profile.
RADIUS	Optional; set Radius, if the attribute and value need to be defined for RADIUS.

Table 2-5 Profile Properties (continued)

Fields	Description
VENDOR	Optional; set Vendor, if the attribute and value need to be defined for Vendor.
DIAMETER	Optional; set Diameter, if the attribute and value need to be defined for Diameter.
Attribute Name	Optional; based on the Attribute Type selected, the attribute name is automated. Set the relevant name for the attribute type selected.
Value Attribute	Optional; set the value for the selected attribute. Click the <b>Add</b> button to save the details and list it in Radius and Value list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.

You can use the Profiles page for the following:

- [Filtering Records](#)
- [Adding Profile Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Profile Details

To add new profile details:

- 
- Step 1** Choose **Configuration > Profiles**. The Profiles page is displayed.
  - Step 2** Click **Add**. The Add Profile page is displayed.
  - Step 3** Specify the required details.
  - Step 4** Click **Submit** to save the specified details in the Profiles page. Otherwise click **Cancel** to return to the Profiles page without saving the details. On successful creation of the profiles, the Profiles page is displayed else a respective error message is displayed.
- 

## UserGroups

The **UserGroups** objects allow you to maintain common authentication and authorization attributes in one location, and then have many users reference them. By having a central location for attributes, you can make modifications in one place instead of having to make individual changes throughout your user community.

For example, you can use several **UserGroups** to separate users by the services they use, such as a group specifying PPP and another for Telnet.

Table 2-6 lists and describes the fields in the Add User Groups page.

**Table 2-6 UserGroups Properties**

Fields	Description
<b>General Properties tab</b>	
UserGroup Name	Required; must be unique in the <b>UserGroup</b> list.
Description	Optional; description of the group.
BaseProfile	Optional; when you set this to the name of a profile, Cisco Prime Access Registrar adds the properties in the Profile to the response dictionary as part of the authorization.
AuthenticationScript	Optional; when you set this property to the name of a script, you can use the Script to perform additional authentication checks to determine whether to accept or reject the user.
AuthorizationScript	Optional; when you set this property to the name of a script, you can use the script to add, delete, or modify the attributes of the Response dictionary.
<b>Attribute List tab</b>	
RADIUS	Optional; set Radius, if the attribute and value need to be defined for RADIUS.
VENDOR	Optional; set Vendor, if the attribute and value need to be defined for Vendor.
DIAMETER	Optional; set Diameter, if the attribute and value need to be defined for Diameter.
Attribute Name	Optional; based on the Attribute Type selected, the attribute name is automated. Set the relevant name for the attribute type selected.
Attribute Value	Optional; set the value for the selected attribute. Click the <b>Add</b> button to save the details and list it in Name and Value list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.
<b>CheckItems List tab</b>	
RADIUS	Optional; set Radius, if the attribute and value need to be defined for RADIUS.
VENDOR	Optional; set Vendor, if the attribute and value need to be defined for Vendor.
DIAMETER	Optional; set Diameter, if the attribute and value need to be defined for Diameter.
Attribute Name	Optional; based on the Attribute Type selected, the attribute name is automated. Set the relevant name for the attribute type selected.
Attribute Value	Optional; set the value for the selected attribute. Click the <b>Add</b> button to save the details and list it in Check Name and Check Value list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.

You can use the User Groups page for the following:

- [Filtering Records](#)
- [Adding UserGroup Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding UserGroup Details

To add new user groups details:

- 
- Step 1** Choose **Configuration > UserGroups**. The User Groups page is displayed.
  - Step 2** Click **Add** to add new user group details. The Add UserGroup page is displayed.
  - Step 3** Specify the required details.
  - Step 4** Click **Submit** to save the specified details in the User Groups page. Otherwise click **Cancel** to return to the User Groups page without saving the details.

On successful creation of the user groups, the User Groups page is displayed else a respective error message is displayed.

---

## UserList

The UserLists object contains all of the individual UserLists, which in turn, contain the specific users stored within Prime Access Registrar. Prime Access Registrar references each specific UserList by name from a Service whose type is set to local. When Prime Access Registrar receives a request, it directs it to a Service. When the Service has its type property set to local, the Service looks up the user's entry in the specific UserList and authenticates and/or authorizes the user against that entry.

You can have more than one UserList in the UserLists object. Therefore, use the UserLists object to divide your user community by organization. For example, you might have separate UserLists objects for Company A and B, or you might have separate UserLists objects for different departments within a company.

Using separate UserLists objects allows you to have the same name in different lists. For example, if your company has three people named Bob and they work in different departments, you could create a UserList for each department, and each Bob could use his own name. Using UserLists lets you avoid the problem of Bob1, Bob2, and so on.

If you have more than one UserList, Prime Access Registrar can run a script in response to requests. The script chooses the Service, and the Service specifies the actual UserList which contains the user. The alternative is dynamic properties.

**Note**

The attributes defined for a user list must match the protocol of the incoming packet. For example, if the incoming packet is a Diameter packet, the attributes defined must be specific to Diameter or common to both RADIUS and Diameter. Similarly, if the incoming packet is a RADIUS packet, the attributes defined must be specific to RADIUS or common to both RADIUS and Diameter. Otherwise, the incoming packet will not be processed.

[Table 2-7](#) lists and describes the fields in the Add User List page.

**Table 2-7 User List Properties**

Fields	Description
UserList Name	Required; must be unique.
Description	Optional; description of the user list.

You can use the User List page for the following:

- [Filtering Records](#)
- [Adding UserList Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding UserList Details

To add new user list details:

- 
- Step 1** Choose **Configuration > UserList**. The User List page is displayed.
- Step 2** Click **Add** to add new user list details. The Add UserList page is displayed.
- Step 3** Enter the required details.
- Step 4** Click **Submit** to save the specified details in the User List page. Otherwise click **Cancel** to return to the User List page without saving the details.

On successful creation of the user list, the User List page is displayed else a respective error message is displayed.

**Note**

After adding a new user list, you can add users to the user list. See [Adding User Details](#) for more information.

## Users

The user objects are created to hold the necessary details to authenticate or authorize a user. These users form the component of User Lists, where their details are stored within Prime Access Registrar. The users in local Userlist can have multiple profiles.



### Note

Username might not include the forward slash (/) character. If the Prime Access Registrar server receives an access request packet with a Username attribute containing a forward slash character and the Prime Access Registrar server uses an internal UserList to look up users, the server produces an error (AX\_EINVAL) and might fail. If usernames require a forward slash, use a script to translate the slash to an acceptable, unused character.

Table 2-8 lists and describes the fields in the Add Users page.

**Table 2-8 Users Properties**

Fields	Description
<b>General Properties tab</b>	
Name	Required; must be unique.
Enabled	Required; must be checked to allow user access. If Enabled is not checked, user is denied access.
Allow Null Pwd	During authentication, if the Allow NULL Password environment variable is set to TRUE, user authentication is bypassed. By default, the Allow NULL Password environment variable is not set.
UserGroup	Use the drop-down list to select a UserGroup and use the properties specified in the UserGroup to authenticate and/or authorize the user. The default is none.
Password	Required; length must be between 0-253 characters.
Base Profile	Optional; use the drop-down list to select a Profile. If the service-type is not equal to Authenticate Only, Prime Access Registrar adds the properties in the Profile to the Response dictionary as part of the authorization. This field is optional for the CLI, but required for the GUI. Use the menu to select a profile other than the default None.
Confirm Password	Required; must match password.
User Defined	Optional; you can use this property to store notational information which you can then use to filter the UserList. This property also sets the environment variable for UserDefined.
Authentication Script	Optional; use the drop-down list to select the name of a script to perform additional authentication checks to determine whether to accept or reject the user. This field is optional for the CLI, but required for the GUI. Use the menu to select an Authentication Script other than the default None.
Authorization Script	Optional; use the drop-down list to select the name of a script to add, delete, or modify the attributes of the Response dictionary. This field is optional for the CLI, but required for the GUI. Use the menu to select an Authorization Script other than the default None.
Description	Optional; description of the user.
<b>Attribute List tab</b>	

**Table 2-8 Users Properties (continued)**

Fields	Description
RADIUS	Optional; set Radius, if the attribute and value need to be defined for RADIUS.
VENDOR	Optional; set Vendor, if the attribute and value need to be defined for Vendor.
Attribute Name	Optional; based on the Attribute Type selected, the attribute name is automated. Set the relevant name for the attribute type selected.
Attribute Value	Optional; set the value for the selected attribute. Click the <b>Add</b> button to save the details and list it in Name and Value list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.
<b>CheckItems List tab</b>	
RADIUS	Optional; set Radius, if the attribute and value need to be defined for RADIUS.
VENDOR	Optional; set Vendor, if the attribute and value need to be defined for Vendor.
Attribute Name	Optional; based on the Attribute Type selected, the attribute name is automated. Set the relevant name for the attribute type selected.
Attribute Value	Optional; set the value for the selected attribute. Click the <b>Add</b> button to save the details and list it in Check Name and Check Value list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.

You can use the Users page for the following:

- [Filtering Records](#)
- [Adding User Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding User Details

To add new user details:

- 
- Step 1** Choose **Configuration > UserList**. The User List page is displayed.
  - Step 2** Click the user list name link. The Users page is displayed.
  - Step 3** Click **Add** to add new user details. The Add Users page is displayed.
  - Step 4** Specify the required details.
  - Step 5** Click **Submit** to save the specified details in the Users page. Otherwise click **Cancel** to return to the Users page without saving the details.
- On successful creation of the user details, the Users page is displayed else a respective error message is displayed.
-

## Scripts

The **Script** objects define the function Cisco Prime Access Registrar invokes whenever the **Script** is referenced by name from other objects in the configuration.

There are four types of scripts:

- REX (RADIUS EXtension) scripts are written in C or C++, and thus are compiled functions that reside in shared libraries
- TCL scripts are written in TCL, and are interpreted functions defined in source files.
- Java scripts
- Internal scripts, which allow you to add, modify, or delete attributes in the request, response, and environment dictionaries for RADIUS, Diameter, and TACACS+. For more information about internal scripts, see the “Using Extension Points” chapter of the *Cisco Prime Access Registrar 8.0 Administrator Guide*.

When you use a Prime Access Registrar file service, Prime Access Registrar automatically closes any opened files. However, if you write scripts that manipulate files, you are responsible for closing them.

If you have more than one extension point script (defined under **/Radius/Scripts**) using the same Java class, only one instance of the class is created and used for all the extension point scripts.

For more information about scripts, see the “Using Extension Points” chapter of the *Cisco Prime Access Registrar 8.0 Administrator Guide*.

Table 2-9 lists and describes the fields in the Add Scripts page.

**Table 2-9 Script Object Properties**

Fields	Description
Script Name	Required; must be unique in the Scripts list.
Language	Required; specify either REX, TCL, Java, or Internal.
Description	Optional; description of the script.
File/Class Name	Required; specifies either a relative or absolute path. When you specify a relative path, the path must be relative to the <b>\$INSTALL/scripts/radius/\$Language</b> directory. When you specify an absolute path, the server must be able to reach it.  For Java language scripts, the name of the class that implements the extension interface; the <b>.class</b> file should be placed in <b>/cisco-ar/scripts/radius/java</b>
Entry Point	Required; when not set, Prime Access Registrar uses the value specified in the <b>Name</b> property.
Init Entry Point	Optional; if set, it must be the name of the global symbol Prime Access Registrar should call when it initializes the shared library at system start up, and just before it unloads the shared library.

Table 2-9 Script Object Properties (continued)

Fields	Description
Init Entry Point Arg	Optional; when set, it provides the arguments to be passed to the <b>InitEntryPoint</b> in the environmental variable <b>Arguments</b> .  <b>Note</b> The <b>InitEntryPoint</b> properties allow you to perform initialization before processing and then cleanup before stopping the server. For example, when Prime Access Registrar unloads the script (when it stops the RADIUS server) it calls the <b>InitEntryPoint</b> again to allow it to perform any clean-up operations as a result of its initialization. One use of the function might be to allow the script to close an open Accounting log file before stopping the RADIUS server.
The following fields appear if the language is set as <b>Internal</b>	
Protocol	Required; select RADIUS or Diameter to indicate the protocol for which the attributes are to be modified.
ActionStatements	Select one of following the options: <ul style="list-style-type: none"> <li>• Simple Attribute Operation—allows you to add, modify, or delete an attribute value to the request, response, or environment dictionary</li> <li>• Copy an Attribute—allows you to copy an attribute value from one dictionary to another</li> <li>• Concatenate Operation—allows you to concatenate an attribute value from one dictionary to another</li> <li>• Replace Operation—allows you to replace an attribute value from one dictionary to another</li> <li>• Value Based Manipulations—allows you to manipulate attribute values in a dictionary based on a given text</li> <li>• Log or Trace Messages—allows you to create different levels of log or trace messages</li> <li>• I can do it myself—allows you to create your own script for the selected protocol</li> </ul>
<b>Left Side of Statement</b>	
Operation	Choose the operation to perform as <b>Add</b> , <b>Modify</b> , or <b>Delete</b> .
Dictionary	Choose <b>Request</b> , <b>Response</b> , or <b>Environment</b> to specify the RADIUS dictionary to apply the action to.
Attr Type	Applicable for RADIUS protocol; select <b>RADIUS</b> or <b>VENDOR</b> to indicate the attribute type.
Group AVP	Applicable for Diameter protocol; select the group AVP and its level to apply the action to.
Attribute	Based on the attribute type/group AVP selected, choose the appropriate attribute to apply the action to.
Env Attribute	Enter the environment attribute to apply the action to.  This field is available only if the Dictionary chosen is <b>Environment</b> .
<b>Right Side of Statement</b>	

File/Class Name	<p>Required; specifies either a relative or absolute path. When you specify a relative path, the path must be relative to the <b>\$INSTALL/scripts/radius/\$Language</b> directory. When you specify an absolute path, the server must be able to reach it.</p> <p>For Java language scripts, the name of the class that implements the extension interface; the <b>.class</b> file should be placed in <b>/cisco-ar/scripts/radius/java</b></p>
Entry Point	<p>Required; when not set, Prime Access Registrar uses the value specified in the <b>Name</b> property.</p>
Init Entry Point	<p>Optional; if set, it must be the name of the global symbol Prime Access Registrar should call when it initializes the shared library at system start up, and just before it unloads the shared library.</p>



**Table 2-9 Script Object Properties (continued)**

<b>Fields</b>	<b>Description</b>
Dictionary	If the type is Radius, choose <b>Request</b> , <b>Response</b> , or <b>Environment</b> to specify the RADIUS dictionary to apply the action to.
Attr Type	Applicable for RADIUS protocol; select <b>RADIUS</b> or <b>VENDOR</b> to indicate the attribute type.
Group AVP	Applicable for Diameter protocol; select the group AVP and its level to apply the action to.
Attribute	Based on the attribute type/group AVP selected, choose the appropriate attribute to apply the action to.
Env Attribute	Enter the environment attribute to apply the action to. This field is available only if the Dictionary chosen is <b>Environment</b> .
<b>Text Manipulations</b>	
This section is available if the Action Statements field is set to <b>Value Based Manipulations</b> .	
Operation	Select one of the following options: <ul style="list-style-type: none"> <li>• <b>Begins With</b>—to manipulate the attribute value beginning with the given text</li> <li>• <b>Contains</b>—to manipulate the attribute value that contains the given text</li> <li>• <b>Ends With</b>—to manipulate the attribute value that ends with the given text</li> <li>• <b>Strip Text</b>—to strip the given text from the attribute value</li> </ul>
Text	The text you need to manipulate the attribute value with.
The following fields are available if the Action Statements field is set to <b>Log or Trace Messages</b> .	
Log Type	Select one of the following options: <ul style="list-style-type: none"> <li>• <b>log</b>—to add a log message</li> <li>• <b>trace</b>—to add a trace message</li> </ul>
Level	Applicable only for logs; level of the log message to add.
Message	The log or trace message to add.
This following field is available if the Action Statements field is set to <b>I can do it myself</b> .	
Statement	Enter the action statement as a free text.

You can use the Scripts page for the following:

- [Filtering Records](#)
- [Adding Script Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Script Details

To add new script details:

- 
- Step 1** Choose **Configuration > Scripts**. The Scripts page is displayed.
  - Step 2** Click **Add** to add new scripts details. The Script Details page is displayed.
  - Step 3** Enter the required details.
  - Step 4** Click **Save** to save the specified details in the Scripts page. Otherwise click **Cancel** to return to the Scripts page without saving the details.
- On successful creation of the scripts, the Scripts page is displayed else a respective error message is displayed.
- 

## Policies

A Policy is a set of rules applied to an Access-Request.

[Table 2-10](#) lists and describes the fields in the Add Policies page.

**Table 2-10** Policies Properties

Fields	Description
Name	Required; must be unique in the <b>Policies</b> list
Description	Optional; description of the Policy
Rules/Policies	Required; set the rules/policies to be grouped.
Operators	Required; set the operators to be grouped along with selected rules/policies. The selected rules and operators will be grouped and listed in the Grouping Box. To delete the available groups, select the relevant group from the Grouping list and click the <b>Delete</b> button below.
Grouping	Optional; grouping of rules.

You can use the Policies page for the following:

- [Filtering Records](#)
- [Adding Policy Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Policy Details

To add new policy details:

- 
- Step 1** Choose **Configuration > Policies**. The Policies page is displayed.

- Step 2** Click **Add** to add new policy details. The Policy Details page is displayed.
- Step 3** Specify the required details.
- Step 4** Click **Submit** to save the specified details in the Policies page. Otherwise click **Cancel** to return to the Policies page without saving the details.

On successful creation of the policies, the Policies page is displayed else a respective error message is displayed.

## GroupServers

Prime Access Registrar allows group-based load balancing among Diameter peers.

### Group-Based Load Balancing

Using this option you can create two or more groups of Diameter remote servers. Each of these groups will have a unique set of remote servers, i.e. no two groups will share the same remote server.

The traffic between each of these groups is load-balanced in failover mode; while traffic between remote servers within the same group is load-balanced based on round-robin or failover mode depending on the Diameter group server properties. The priority of each of the groups is set with the help of metrics.

The workflow for group-based load balancing is as given below:

1. Traffic from Prime Access Registrar to a remote server, via Diameter proxy service, is directed through the first group, till Prime Access Registrar has active communication channel with at least one remote server belonging to the first group.
2. When Prime Access Registrar loses connectivity with all the remote servers in the first group, it directs the rest of the Diameter traffic towards remote servers belonging to the second group.

**Table 2-11** *Diameter GroupServer Properties*

Fields	Description
<b>General Properties tab</b>	
Name	Required; name of the group server.
MultiplePeersPolicy	Required; Policy used by the Prime Access Registrar server to load balance the peers within the group. This could be one of the following: <ul style="list-style-type: none"> <li>• FailOver—Traffic is directed towards first priority remote server within the group. When Prime Access Registrar loses connectivity with the first priority remote server, it directs the subsequent traffic towards the second priority remote server within the group.</li> <li>• RoundRobin—Traffic is distributed across all the active remote servers within the group.</li> </ul>
GroupTimeOutPolicy	Required; action to perform when there is a timeout with the group server. This could be FailOver, DropPacket, or SendError.
<b>DiameterRemoteServersList</b>	
List of Diameter remote servers to add to the group.	
Name	Required; name of the peer.

**Table 2-11 Diameter GroupServer Properties (continued)**

Fields	Description
Metric	Required; metric value for this peer entry. The higher the value the lower the preference. The highest value of preference is 0.
Weight	Required; default value is 0. Specifies the weight percentage for which the server group needs to load balance the peer.  <b>Note</b> When you set the weight to a value other than 0, the weight should be in multiples of 10 and the sum of the weights configured in the peer list should be equal to 100.
IsActive	Optional; if this is checked, the new sessions will not go to the peer server. By default, this is unchecked.

You can use the GroupServers page for the following:

- [Filtering Records](#)
- [Adding Group Server Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Group Server Details

To add new group servers:

- 
- Step 1** Choose **Configuration > GroupServers**. The GroupServers page is displayed.
  - Step 2** Click **Add** to add new group server details. The Group Servers page is displayed.
  - Step 3** Specify the required details.
  - Step 4** Click **Save GroupServer** to save the specified details in the Group Servers page. Otherwise click **Cancel** to return to the GroupServers page without saving the details.

On successful creation of the group server, the GroupServers page is displayed else a respective error message is displayed.

---

## Services

Cisco Prime Access Registrar supports authentication, authorization, and accounting (AAA) services. In addition to the variety of built-in AAA services (specified in the **Type** property), Cisco Prime Access Registrar also enables you to add new AAA services through custom shared libraries.

This section lists the types of services available in Prime Access Registrar with their required and optional properties. The service you specify determines what additional information you must provide. The various types of services are:

- [Simple Services](#)

- [ServiceWithRS](#)
- [PEAP Service](#)
- [EAP Service](#)
- [Diameter Service](#)

## Simple Services

Prime Access Registrar provides the following simple services:

- [Rex](#)
- [File](#)
- [Trusted-ID](#)
- [Group](#)
- [Local](#)
- [Java](#)
- [WiMAX](#)
- [RADIUS-Query](#)
- [Dyn-Authz](#)
- [Diameter-RADIUS](#)
- [RADIUS-Diameter](#)
- [Diameter-Query](#)
- [3GPPAuthorization](#)
- [3GPP-Reverse-Authorization](#)

### Rex

Select rex service when a custom service needs to be created and a script for authentication, authorization, or accounting has to be used.

### File

Select File type when local accounting is to be performed using a specific file. The files under the configuration will be saved in the configured name when the server is invoked even if the service is not being invoked by any request packets.

Prime Access Registrar flushes the accounting record to disk before it acknowledges the request packets. Based on the specified maximum file size and age, it closes the accounting file, moves it to a new name, and reopens the file as a new file. The file names are based on its creation and modification dates.

### Trusted-ID

Select the trusted-id service type to authorize and authenticate a user based on a Trusted ID. Using SSG's Transparent Auto-Login (TAL) feature, a TAL access-request packet contains a Trusted ID, such as a MAC address, that identifies the user without the user's real username and password. If Prime Access Registrar knows the user associated with the Trusted ID, it uses the Trusted ID to authenticate and authorize the user. For more information, see the “Using Trusted ID Authorization with SESM” chapter of the [Cisco Prime Access Registrar 8.0 Administrator Guide](#).

## Group

A group service contains a list of references to other services and specifies whether the responses from each of the services should be handled as a logical AND or OR function, which is specified in the Result-Rule attribute of Group Services. The default value is AND.

When the Result-Rule attribute is set to AND or OR, each referenced service is accessed sequentially, and the Group Service waits for a response from the first referenced service before moving on to the next service (if necessary).

The ResultRule settings parallel-and and parallel-or are similar to the AND and OR settings except that they ask each referenced service to process the request simultaneously instead of asking each referenced server sequentially, thereby saving processing time.

## Local

Select local services when authentication and authorization needs to be performed by Prime Access Registrar server using a specific UserList.

## Java

Select Java service type when a custom service needs to be created and to use an extension point script to provide the service's functionality and handle both RADIUS and TACACS requests for authentication, authorization, or accounting.

## WiMAX

Prime Access Registrar uses the Extensible Authentication Protocol (EAP) to enable the WiMAX feature. It captures the IP attributes and Mobility Keys that are generated during network access authentication.

## RADIUS-Query

Select this service type to query cached data through RADIUS Packets. It contains the list of session managers to be queried from and a list of (cached) attributes to be returned in the Access-Accept packet in response to a RADIUS Query request. It is initiated through an extension point script or through the Rule and Policy Engine by setting it to a new environment variable named Query-Service.

## Dyn-Authz

Select this service type to process dynamic authorization requests. This involves Change of Authorization (COA) and Packet of Disconnect (POD) features. For more information about these features, see [Chapter 9, "Using Cisco Prime Access Registrar Server Features."](#)

## Diameter-RADIUS

Select this service for Diameter to RADIUS translation to translate incoming Diameter request to a RADIUS equivalent and then the RADIUS response to Diameter equivalent. Prime Access Registrar provides scripting points, which operate on the original packet and on the newly translated packet based on request and response mapping. For more information, see [Chapter 4, "Diameter."](#)

## RADIUS-Diameter

Select this service for RADIUS to Diameter translation to translate incoming RADIUS request to a Diameter equivalent and then the Diameter response to RADIUS equivalent. Prime Access Registrar provides scripting points, which operate on the original packet and on the newly translated packet based on request and response mapping. For more information, see [Chapter 4, “Diameter.”](#)

## Diameter-Query

Select this service type to query cached data through Diameter Packets. It contains the list of session managers to be queried from and a list of (cached) attributes to be returned in the Access-Accept packet in response to a Diameter Query request. It is initiated through an extension point script or through the Rule and Policy Engine by setting it to a new environment variable named Query-Service.

## 3GPPAuthorization

Select this service to enable 3GPP authorization of subscribers. For more information about 3GPP authorization, see the “Wireless Support” chapter of the [Cisco Prime Access Registrar 8.0 Reference Guide](#).

## 3GPP-Reverse-Authorization

Select this service to enable 3GPP reverse authorization of subscribers. For more information about 3GPP reverse authorization, see the “Wireless Support” chapter of the [Cisco Prime Access Registrar 8.0 Reference Guide](#).

[Table 2-12](#) lists and describes the fields in the Services Details page. The fields listed below are the entire list of all the available types. The fields are displayed based on the type selected.

**Table 2-12 Simple Service Properties**

Fields	Description
Service Name	Required; must be unique in the Services list.
Incoming Script	Optional; name of script to run when the service starts.
Type	Required; must set it to a valid Prime Access Registrar service.
Outgoing Script	Name of script to run when the service ends.
Description	Optional; description of the service.
Outage Script	Optional; if you set this property to the name of a script, Cisco Prime Access Registrar runs it when an outage occurs. This property allows you to create a script that notifies you when the RADIUS server detects a failure.
Outage Policy	Required; the default is <b>DropPacket</b> . This property defines how Cisco Prime Access Registrar handles requests if all servers listed in the <b>RemoteServers</b> properties are unavailable (that is, all remote RADIUS servers are not available). You must set it to one of the following: <b>AcceptAll</b> , <b>DropPacket</b> , or <b>RejectAll</b> .

The following properties appear for the job type **rex**.

Table 2-12 Simple Service Properties (continued)

Fields	Description
Filename	Required; must be either a relative or an absolute path to the shared library containing the Service. When the pathname is relative, it must be relative to <b>\$INSTALL/Scripts/Radius/rex</b> .
EntryPoint	Required; must be set to the function's global symbol.
InitEntryPoint	Required; must be the name of the global symbol Cisco Prime Access Registrar should call when it initializes the shared library and just before it unloads the shared library.  A rex service must have an InitEntryPoint even if the service only returns REX_OK.
InitEntryPointArgs	Optional; when set, it provides the arguments to be passed to the <b>InitEntryPoint</b> in the environmental variable <b>Arguments</b> .

The following properties appear for the job type **file**.

FilenamePrefix	Required; a string that specifies where Cisco Prime Access Registrar writes the account records. It must be either a relative or absolute path. When you specify a relative path, it must be relative to the <b>\$INSTALL/logs</b> directory. When you specify an absolute path, the server must be able to reach it. The default is <b>Accounting</b> .
MaxFileAge	Optional; stored as a string, but is composed of two parts, a number and a units indicator (<n> <units>) in which the unit is one of: H, Hour, Hours, D, Day, Days, W, Week, Weeks. The default is one day.
RolloverSchedule	Indicates the exact time including the day of the month or day of the week, hour and minute to roll over the accounting log file.
MaxFileSize	Optional; stored as a string, but is composed of two parts, a number and a units indicator (<n> <units>) in which the unit is one of: K, kilobyte, or kilobytes, M, megabyte, or megabytes, or G, gigabyte, or gigabytes. The default is ten megabytes.
UseLocalTimeZone	When set to TRUE, indicates the accounting records' TimeStamp is in local time. When set to FALSE, the default, accounting records' TimeStamp is in GMT.
FileType	Choose <b>log</b> or <b>csv</b> to indicate the file type to export the accounting records to. If you choose <b>log</b> , the Prime Access Registrar server writes accounting messages to the <b>accounting.log</b> file in the <b>/opt/CSCOar/logs</b> directory. If you choose <b>csv</b> , the Prime Access Registrar server writes accounting messages to the <b>accounting.csv</b> file in the <b>/opt/CSCOar/logs</b> directory.
EnableRollOverIntelligence	Check the box to enable rollover intelligence for the accounting records based on the accounting service properties. For more information, see <a href="#">Rolling Encryption Support for Pseudonym Generation in EAP-SIM, EAP-AKA, and EAP-AKA' Services</a> , page 5-61.
AttributesToBeLogged	The selected list of attributes that must be logged. If the list is empty, the accounting file service logs all the attributes of the packet.
Delimiter	The delimiter to use in the accounting file. This field is available if you set the FileType as <b>csv</b> . Delimiters could be ';', ',', and ':' and default value is ','.

The following properties appear for the job type **trusted-id**.

**Table 2-12 Simple Service Properties (continued)**

Fields	Description
UserService	Required; name of service that can be used to authenticate.
SessionManager	Required; select the required session manager from the available list.
The following properties appear for the job type <b>group</b> .	
Result Rule	<p>When set to AND (the default), the response from the GroupService is positive if each of the services referenced return a positive result. The response is negative if any of the services reference return a negative result.</p> <p>When set to OR, the response from the GroupService is positive if any of the services referenced return a positive result. The response is negative if all the referenced services return a negative result.</p> <p>The settings parallel-AND or parallel-OR are similar to AND and OR settings, except that each referenced service processes requests simultaneously instead of asking each reference service sequentially to save processing time.</p>
GroupServices	<p>Optional; use the GroupServices subdirectory to specify the subservices in an indexed list to provide specific ordering control of which services to apply first. Each subservice listed must be defined in the Services section of the RADIUS configuration and cannot be a of type group, eap-leap, or eap-md5.</p> <p>To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.</p>
The following properties appear for the job type <b>local</b> .	
UserList	<p>Required; this object contains all of the individual UserLists, which in turn, contain the specific users stored within Prime Access Registrar. Cisco Prime Access Registrar references each specific UserList by <b>name</b> from a Service whose type is set to <b>local</b>.</p> <p>When Cisco Prime Access Registrar receives a request, it directs it to a Service. When the Service has its type property set to <b>local</b>, the Service looks up the user's entry in the specific UserList and authenticates and/or authorizes the user against that entry.</p>
Enable Device Access	<p>Check the box to enable TACACS+ command authorization.</p> <p><b>Note</b> Device Access Rules are applicable for TACACS+ command authorization. For more information, see <a href="#">TACACS+ Support for AAA, page 9-57</a>.</p>
Device Access Rule	Select a device access rule and click <b>Add</b> . The selected access rule is displayed in the Device Access Rules list box.
Default Device Access Action	Select the default action to perform on the commands for all the access rules in the authorization service. Options are <b>PermitAll</b> and <b>DenyAll</b> .
The following properties appear for the job type <b>java</b> .	
Class name	Optional; set to the name of a class that implements the Extension interface.
InitializeArg	Optional; set to a string to be passed to the Initialize method if the class implements the optional ExtensionWithInitialization interface.
The following properties appear for the job type <b>wimax</b> .	

**Table 2-12 Simple Service Properties (continued)**

Fields	Description
HARKKey	Required; used as the base key to generate random HARKKey for all the HAs that are configured in Prime Access Registrar. By default, the value is <code>cisco112</code> . You can change this value.
WimaxAuthenticationService	Required; a valid EAP service which can be used for WiMAX authentication. By default, this value is none.
HARKLifetime	Required; used as time (in minutes) to regenerate the HARKKeys based on its lifetime.
WimaxSessionManager	Required; set a valid session manager which has HA and HA Cache as resource managers. By default, this value is none.
WimaxQueryService	Required; set a valid RADIUS query service which is configured with WiMAX session manager. By default, this value is none.
WimaxPrepaidService	Optional; set a valid prepaid service to carry out the prepaid functionality of WiMAX. Otherwise this value is set to none.
AllowAAAToIncludeKeys	Optional; If this is set, the HAAA will include the hHA-RK-Key, hHA-RK-SPI and hHA-RK-Lifetime in the Access-Accept. Otherwise, those attributes will not be in the Access-Accept. By default this value is True.
RequiredMSK	Optional; If this is set, the MSK will be provided by the AAA server as a result of successful EAP-Authentication. By default, this value is False.

The following properties appear for the job type **radius-query**.

#### Attribute List tab

Attribute type	Select either <b>RADIUS</b> or <b>VENDOR</b> . If Vendor is selected, specify the vendor type from the drop-down list. Select the attributes from the available list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.
----------------	---

#### Session Manager tab

Session Manager	Select the required session manager from the available list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.
-----------------	--

The following property appears for the job type **dyn-auth**.

Session Cache Query Service	Select the session cache query service to use for dynamic authorization.
-----------------------------	--

The following properties appear for the job type **diameter-radius** or **radius-diameter**.

ProxyServiceName	Select the Diameter proxy service name.
DiameterApplicationID	Select the Diameter service application ID. This field appears only for radius-diameter service type.
EnableRequestCommandMappings	Check this box to enable command mapping.

Table 2-12 Simple Service Properties (continued)

Fields	Description
SendRAR-ASRToClient	Check the box if the COA/POD packets received by Prime Access Registrar are to be translated and sent as Re-Auth-Request (RAR) / Abort Session Request (ASR) to a Diameter client. This field appears only for radius-diameter service type.
ClientHostName	Hostname of the Diameter client to which the translated RAR/ASR must be sent. If the session manager is configured, the client host name can be acquired from it using the Session-Manager AVP.  This field appears only for radius-diameter service type.
UseFor3GPPReverseAutho- rizationService	Check the box to enable 3GPP authorization service in the translation framework. This field appears only for radius-diameter service type.
PreRequestTranslationS- cript	Select the scripting point to be called on the original request packet.
PostRequestTranslation Script	Select the scripting point to be called on the translated request packet.
PreResponseTranslation Script	Select the scripting point to be called on the response packet.
PostResponseTranslatio- nScript	Select the scripting point to be called on the translated response packet.
CommandMappings	This tab allows you to map commands.
ResultCodeMappings	This tab allows you to map result codes.
RequestAVPMappings	This tab allows you to map request AVPs.
RequestAVPsToBeAdde- d	This tab allows you to map request AVPs to be added.
RequestEnvironmentMa- ppings	This tab allows you to map request environment variables.
ResponseAVPMappings	This tab allows you to map response AVPs.
ResponseAVPsToBeAd- ded	This tab allows you to map response AVPs to be added.
ResponseEnvironment Mappings	This tab allows you to map response environment variables.
The following properties appear for the job type <b>diameter-query</b> .	
UpdateSessionLastAcce- ssTime	Check the box to update the timestamp when the Diameter session was last accessed or called.
<b>Attribute List tab</b>	
Attribute type	Select either <b>RADIUS</b> or <b>VENDOR</b> . If Vendor is selected, specify the vendor type from the drop-down list. Select the attributes from the available list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.
<b>Session Manager tab</b>	
Session Manager	Select the required session manager from the available list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.

**Table 2-12 Simple Service Properties (continued)**

Fields	Description
The following property appears for the job type <b>3gpp-authorization</b> .	
Protocol	Required; select RADIUS or Diameter to indicate the protocol to use for 3GPP authorization.
FetchLocationInformation	Check the box to fetch location related information of the RADIUS/Diameter client for the 3GPP authorization service.  Prime Access Registrar allows or blocks access of a subscriber to voice over Wi-Fi (VoWiFi) based on the location information. For more details on voice over Wi-Fi (VoWiFi) location-based authentication, see the “Wireless Support” chapter of the <i>Cisco Prime Access Registrar 8.0 Reference Guide</i>
TranslationService	Required if the protocol selected is RADIUS; translation service to use during 3GPP authorization.
DiameterProxyService	Required if the protocol selected in Diameter; diameter proxy service to use during 3GPP authorization.
The following properties appear for the job type <b>3gpp-reverse-authorization</b> .	
TranslationService	Required; the translation service to use for 3GPP reverse authorization.

You can use the Simple Services List page for the following:

- [Filtering Records](#)
- [Adding Simple Service Details](#)
- [Editing Records](#)
- [Deleting Records](#)

### Adding Simple Service Details

To add new simple service details:

- 
- Step 1** Choose **Configuration > Services > Simple**. The Services List(REX, FILE, LOCAL, GROUP, JAVA...) page is displayed.
  - Step 2** Click **Add** to add new simple service details. The Services Details page is displayed.
  - Step 3** Enter the required details.
  - Step 4** Click **Submit** to save the specified details in the Services List(REX, FILE, LOCAL, GROUP, JAVA...) page. Otherwise click **Cancel** to return to the Services List(REX, FILE, LOCAL, GROUP, JAVA...) page without saving the details.

On successful creation of the simple service properties, the Services List(REX, FILE, LOCAL, GROUP, JAVA...) page is displayed else a respective error message is displayed.

---

## ServiceWithRS

The RemoteServers directory lists one or more remote servers to process access requests. The servers must also be listed in order under /Radius/RemoteServers. The order of the RemoteServers list determines the sequence for directing access requests when MultipleServersPolicy is set to RoundRobin mode. The first server in the list receives all access requests when MultipleServersPolicy is set to Failover mode.

The RemoteServers object can be used to specify the properties of the remote servers to which Services proxy requests. RemoteServers are referenced by name from the RemoteServers list in either the RADIUS, LDAP or TACACS-UDP Services.

Table 2-13 lists and describes the fields in the Services Details page.

**Table 2-13 Remote Server Service Properties**

Fields	Description
Service Name	Required; name of the remote server service
Incoming Script	Optional; name of script to run when the service starts
Type	Required; Remote service Type must be set to one of the following: <b>ldap</b> , <b>ldap-accounting</b> , <b>odbc-accounting</b> , <b>odbc</b> , <b>oci-accounting</b> , <b>oci</b> , <b>prepaid</b> , <b>radius</b> , <b>radius-session</b> , <b>m3ua</b> , or <b>extended-eap</b> .
Outgoing Script	Optional; name of script to run when the service ends.
Outage Script	Optional; if you set this property to the name of a script, Prime Access Registrar runs it when an outage occurs. This property allows you to create a script that notifies you when the RADIUS server detects a failure.
Outage Policy	The default is <b>DropPacket</b> . This property defines how Prime Access Registrar handles requests if all servers listed in the <b>RemoteServers</b> properties are unavailable (that is, all remote RADIUS servers are not available). You must set it to one of the following: <b>AcceptAll</b> , <b>DropPacket</b> , or <b>RejectAll</b> .
Description (optional)	Optional; description of the remote server service
MultipleServersPolicy	Required; must be set to either <b>Failover</b> or <b>RoundRobin</b> .  When you set it to <b>Failover</b> , Prime Access Registrar directs requests to the first server in the list until it determines the server is offline. At which time, Prime Access Registrar redirects all requests to the next server in the list until it finds a server that is online.  When you set it to <b>RoundRobin</b> , Prime Access Registrar directs each request to the next server in the RemoteServers list to share the resource load across all of the servers listed in the RemoteServers list.
NASIDList	Mandatory for extended-EAP service. Select a valid user list as configured under RADIUS > UserLists.  Extended-EAP is used as an authorization service to retrieve authorization information from the remote web server using the REST interface. To configure a REST remote server for extended-EAP service, see <a href="#">REST, page 2-139</a>
RemoteServers	Select the required remote server from the available list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.

**Table 2-13 Remote Server Service Properties (continued)**

Fields	Description
AuthorizationInfo LookUp	Applicable only for the m3ua service type. Choose one of the following from the drop-down list: <ul style="list-style-type: none"> <li>MSISDN-IMSI—To fetch MSISDN in the request and send IMSI in the response to the HLR.</li> <li>IMSI-MSISDN—To fetch IMSI in the request and send MSISDN in the response to the HLR.</li> <li>MAP-RESTORE—To fetch the profile information of a subscriber from the HLR. For more information on configuring the M3UA service with Map Restore Data authorization, see <a href="#">Configuring M3UA Service with Map Restore Data Authorization, page 14-14</a>.</li> </ul>
MapVersion	Applicable only for the m3ua service type; select the map version that HLR supports.

**Device Access Rules**

This section is applicable for TACACS+ command authorization and is available only for service types local-user, oci, odbc, and ldap. For more information on TACACS+ command authorization, see [TACACS+ Support for AAA, page 9-57](#).

Enable Device Access	Check the box to enable TACACS+ command authorization.
Device Access Rule	Select a device access rule and click <b>Add</b> . The selected access rule is displayed in the Device Access Rules list box.
Default Device Access Action	Select the default action to perform on the commands for all the access rules in the authorization service. Options are <b>PermitAll</b> and <b>DenyAll</b> .

**Restore Data Mappings Section**

IMSI	IMSI received in the response from HLR.
Naea-Preferred CI	North American Equal Access preferred Carrier ID List. A list of the preferred carrier identity codes that are subscribed to.
Roaming Restricted In Sgsn Due To Unsupported Feature	Indicates that a subscriber is not allowed to roam in the current Service GPRS Support Node (SGSN) or Cisco Mobility Management Entity (MME) area.
Network Access Mode	The Network Access Mode (NAM) defines if the subscriber is registered to get access to the CS (non-GPRS/EPS network), to the PS (GPRS/EPS) network, or to both networks. NAM describes the first level of the subscriber data pseudo-tree below the IMSIroot. It is permanent subscriber data stored in the HSS / HLR and the SGSN with the Gs interface option, and the MME with the SGs interface option.
LMU Indicator	Indicates the presence of an LMU.
IST Alert Timer	Indicates the IST alert timer value that must be used in the Mobile Switching Center (MSC) to inform the HLR about the call activities that the subscriber performs.
Super Charger Supported In HLR	Indicates whether super charger concept is supported in HLR.

**Table 2-13 Remote Server Service Properties (continued)**

<b>Fields</b>	<b>Description</b>
CS Allocation Retention Priority	Allocation-retention priority for Circuit Switched (CS). This parameter specifies relative importance to compare with other bearers about allocation and retention of bearer.
ChargingCharacteristics	Subscribed charging characteristics.
Access Restriction Data	Allowed Recipient Access Table (RAT) according to subscription data.
UE Reachability Request Indicator	Indicates that the Home Subscriber Server (HSS) is awaiting a notification of user equipment (UE) reachability.
Category	Calling party category
LSA Information	These parameters refer to one or more localized service areas (LSAs) a subscriber may be a member of, together with the priority, the preferential access indicator, the active mode support indicator and active mode indication of each localized service area. The access right outside these localized service areas is also indicated.
<b>Subscriber Data</b>	
MSISDN	MSISDN value in the subscriber data.
Subscriber Status	Barring status of the subscriber, which could be Service Granted or Operator Determined Barring.
Roaming Restriction Due To Unsupported Feature	Indicates that the subscriber is not allowed to roam in the current MSC area.
Bearer Service List	List of extensible bearer services subscribed. Configure the index value to fetch only the required bearer services.
TeleService List	List of extensible teleservices subscribed. Configure the index value to fetch only the required teleservices.
Provisioned SS	List of supplementary services provisioned. Configure the index value to fetch only the required supplementary services.
ODB-Data	Operator Determined Barring (ODB) general data and ODB Home Public Land Mobile Network (HPLMN) specific data.
Regional Subscription Data	List of regional subscription areas (zones) in which the subscriber is allowed to roam. Configure the index value to fetch only the required zones.
VBS Subscription Data	List of Voice Broadcast Services (VBS) subscribed. Configure the index value to fetch only the required VBS.
VGCS Subscription Data	List of Voice Group Call Services (VGCS) subscribed. Configure the index value to fetch only the required VGCS.
<b>LCS Information</b>	
Live Communication Server (LCS) related information for the subscriber.	

**Table 2-13 Remote Server Service Properties (continued)**

Fields	Description
GMLC-List	List of Gateway Mobile Location Centers (GMLCs) that are permitted to issue a call/session unrelated or call/session related MT-LR request. Configure the index value to fetch only the required GMLCs.
LCS-Privacy Exception List	Classes of LCS client that are allowed to locate any target Mobile Station (MS). Configure the index value to fetch only the required classes.
MOLR-List	Code and status of Mobile Originating Location Request (MO-LR) subscribed. Configure the index value to fetch only the required requests.
<b>MC-SS-Info</b> Parameters identifying Multicall (MC) supplementary services (SS) that are subscribed.	
MC-SS-Code	Code of the MC SS.
MC-SS-Status	Status of the MC SS.
NbrSB	Maximum number of parallel bearers that may be used as defined by the user's subscription.
NbrUser	Maximum number of parallel bearers that may be used as defined by the user at registration of the MC SS.
<b>SGSN-CAMEL-Subscription Info</b> Parameters identifying the subscribers as having Customized Application for Mobile Enhanced Logic (CAMEL) services that are invoked in the SGSN.	
GPRS-CSI	Identifies the subscriber as having GPRS originating SMS CAMEL services.
MO-SMS-CSI	Identifies the subscriber as having mobile originating SMS CAMEL services.
MT-SMS-CSI	Identifies the subscriber as having mobile terminating SMS CAMEL services.
<b>ProfileMappings</b>	
Attribute	Select an RADIUS attribute to map the fetched profile data.
Value:Profile	Enter a value for the attribute.
ProfileList	Select one of the profile lists and click <b>Add</b> . The entered profile details are displayed in the list box in the ProfileMappings section. You can delete a profile attribute from the list as required.

You can use the ServiceWithRS List page for the following:

- [Filtering Records](#)
- [Adding Remote Server Service Details](#)
- [Editing Records](#)
- [Deleting Records](#)

### Adding Remote Server Service Details

To add new remote server service details:

- Step 1** Choose **Configuration > Services > ServiceWithRS**. The Services List (..with Remote Servers) page is displayed.

- Step 2** Click **Add** to add new remote server service details. The Services Details page is displayed.
- Step 3** Enter the required details.
- Step 4** Click **Submit** to save the specified details in the Services List (..with Remote Servers) page. Otherwise, click **Cancel** to return to the Services List (..with Remote Servers) List page without saving the details.
- On successful creation of the properties, the Services List (..with Remote Servers) page is displayed else a respective error message is displayed.
- 

## PEAP Service

Protected EAP (PEAP) is an authentication method designed to mitigate several weaknesses of EAP. PEAP leverages Industry standard authentication of the server using certificates TLS (RFC 2246) and creation of a secure session that can then be used to authenticate the client.

The PEAP protocol consists of two phases, an authentication handshake phase and a tunnel phase where another complete EAP authentication exchange takes place protected by the session keys negotiated by phase one. Prime Access Registrar supports the tunneling of other EAP methods within the PEAP phase two exchange.

Prime Access Registrar supports the two major existing variants of PEAP:

- [PEAP Version 0](#) (Microsoft PEAP)
- [PEAP Version 1](#) (Cisco Prime PEAP)

### PEAP Version 0

PEAP Version 0 also called as Microsoft PEAP is described in IETF drafts (draft-kamath-pppext-peapv0-00.txt and draft-josefsson-pppext-eap-tls-eap-02.txt). This version of PEAP uses either EAP-MSChapV2 or EAP-SIM as an authentication method. The testing method used for this version of PEAP is radclient.

### PEAP Version 1

PEAP Version 1 also called as Cisco Prime PEAP is described by IETF draft (draft-zhou-pppext-peapy1-00.txt). This version can use either EAP-GTC or EAP-SIM as an authentication method. The testing method used for this version of PEAP is radclient.

[Table 2-14](#) lists and describes the fields in the PEAP Services Details page. The fields listed below are the entire list of all the available types. The fields are displayed based on the type selected.

**Table 2-14** PEAP Service Properties

Fields	Description
Service Name	Required; service name
Incoming Script	Optional; script Prime Access Registrar server runs when it receives a request from a client.
Type	Required; must set it to a valid Prime Access Registrar service.
Outgoing Script	Optional; script Prime Access Registrar server runs before it sends a response to a client.

**Table 2-14 PEAP Service Properties (continued)**

Fields	Description
Maximum Message Size	Indicates the maximum length in bytes that a PEAP or EAP-TLS message can have before it is fragmented.
Server Certificate File	<p>Required; the full pathname of the file containing the server's certificate or certificate chain used during the TLS exchange. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The two valid encoding prefixes are PEM and DER. If an encoding prefix is not present, the file is assumed to be in PEM format.</p> <p>The following example assumes that the subdirectory <b>pki</b> under <b>/cisco-ar</b> contains the server's certificate file. The file <b>server-cert.pem</b> is assumed to be in PEM format; note that the file extension <b>.pem</b> is not significant.</p> <p><b>set ServerCertificateFile PEM:/cisco-ar/pki/server-cert.pem</b></p>
Private Key Password	Required; the password used to protect the server's private key.
Server RSA Key File	Required; the full pathname of the file containing the server's RSA private key.
CRL Distribution URL	<p>Optional; The URL that Prime Access Registrar should use to retrieve the CRL. You can specify a URL that uses HTTP or LDAP.</p> <p>The following is an example for an HTTP URL: &lt;http://crl.verisign.com/pca1.1.1.crl&gt;.</p> <p>The following is an example for an LDAP URL: ldap://209.165.200.225:388/CN=development-CA,CN=acs-westcoast2,CN=CDP,CN=Public Key Services,CN=Services,CN=Configuration,DC=cisco,DC=com</p>
CA Certificate File	Optional; the full pathname of the file containing trusted CA certificates used for client verification. The file can contain more than one certificate, but all certificates must be in PEM format. DER encoding is not allowed.
Certificate Verification Mode	<p>Optional; specifies the type of verification used for client certificates. Must be set to one of RequireCertificate, None, or Optional.</p> <ul style="list-style-type: none"> <li>RequireCertificate causes the server to request a client certificate and authentication fails if the client refuses to provide one.</li> <li>None will not request a client certificate.</li> </ul> <p>Optional causes the server to request a client certificate but the client is allowed to refuse to provide one.</p>

Table 2-14 PEAP Service Properties (continued)

Fields	Description
CA Certificate Path	<p>Optional; the name of a directory containing trusted CA certificates (in PEM format) used for client verification. This parameter is optional, and if it is used there are some special preparations required for the directory it references.</p> <p>Each certificate file in this directory must contain exactly one certificate in PEM format. The server looks up the certificate files using the MD5 hash value of the certificate's subject name as a key. The directory must therefore also contain a set of symbolic links each of which points to an actual certificate file. The name of each symbolic link is the hash of the subject name of the certificate.</p> <p>For example, if a certificate file name <b>ca-cert.pem</b> is located in the <b>CACertificatePath</b> directory, and the MD5 hash of the subject name contained in <b>ca-cert.path.pem</b> is 1b96dd93, then a symbolic link named 1b96dd93 must point to the <b>ca-cert.pem</b> file.</p> <p>If there are subject name collisions such as multiple certificates with the same subject name, each link name must be indexed with a numeric extension as in 1b96dd93.0 and 1b96dd93.1.</p>
Verification Depth	Optional; specifies the maximum length of the certificate chain used for client verification.
Enable Session Cache	Optional; specifies whether TLS session caching (fast reconnect) is enabled or not. Set to True to enable session caching; otherwise set to False.
Tunnel Service	Required; must be the name of an existing EAP-MSCHAPv2 or EAP-SIM service.
Authentication Timeout	Required; specifies time (in seconds) to wait before an authentication request times out; defaults to 120.
Description (optional)	Optional; description of the PEAP service.
Session Timeout	<p>Optional; if TLS session caching (fast reconnect) is enabled, SessionTimeout specifies the maximum lifetime of a TLS session. Expired sessions are removed from the cache and will require a subsequent full authentication.</p> <p>SessionTimeout is specified as a string consisting of pairs of numbers and units, where units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks, as in the following:</p> <p><b>Set SessionTimeout "1 Hour 45 Minutes"</b></p>
Use ECC Certificates	<p>Check this box, to use the ECC, RSA, or combination of both the certificates for certificate based verification.</p> <p>When this field is disabled, only RSA is used for certificate based verification. The default location to fetch the certificate file is <b>/cisco-ar/pki</b>.</p>
Enable Auto Chaining	When set to TRUE, Prime Access Registrar sends its server certificate chain (Server-Cert -> IntermediateCA -> RootCA) while presenting the server certificate to the client for server side authentication. When set to FALSE, Prime Access Registrar sends only the server certificate (Server-Cert) to the client.
Enable WPS	Optional; When set to TRUE, enables Windows Provisioning Service (WPS) and provides two other properties, MasterURL and WPSGuestUserProfile. The default value is FALSE.

**Table 2-14 PEAP Service Properties (continued)**

Fields	Description
Master URL	Optional; when using WPS, specifies the URL of the provisioning server which is modified with the appropriate fragment and sent to the client.
WPS Guest User Profile	Optional; when using WPS, specifies a profile to be used as a guest user profile; must be a valid profile under <b>/Radius/Profiles</b> .  This profile is used for guests and users whose account has expired. This profile normally contains attributes denoting the VLAN-id of the guest network (which has the provisioning server alone) and might contain IP-Filters that would restrict the access of the guest (to only the provisioning server).

You can use the PEAP Services List page for the following:

- [Filtering Records](#)
- [Adding PEAP Service Details](#)
- [Editing Records](#)
- [Deleting Records](#)

### Adding PEAP Service Details

To add new PEAP service details:

- 
- Step 1** Choose **Configuration > Services > PEAP**. The PEAP Services List page is displayed.
  - Step 2** Click **Add** to add new PEAP service details. The PEAP Services Details page is displayed.
  - Step 3** Specify the relevant PEAP service details.
  - Step 4** Click **Submit** to save the specified details in the PEAP Services List page. Otherwise click **Cancel** to return to the PEAP Services List page without saving the details.

On successful creation of the PEAP service properties, the PEAP Services List page is displayed else a respective error message is displayed.

---

## EAP Service

Prime Access Registrar supports the Extensible Authentication Protocol (EAP) to provide a common protocol for differing authentication mechanisms. It provides dynamic selection of the authentication mechanism at the time of authentication based on information transmitted in the Access-Request.

Prime Access Registrar supports the following EAP authentication methods:

- [EAP-AKA](#)
- [EAP-AKA-Prime](#)
- [EAP-FAST](#)
- [EAP-GTC](#)
- [EAP-LEAP](#)

- EAP-MD5
- EAP-Negotiate
- EAP-MSChapV2
- EAP-SIM
- EAP-Transport Level Security (TLS)
- EAP-TTLS

### EAP-AKA

Authentication and Key Agreement (AKA) is an EAP mechanism for authentication and session key distribution. It is used in the 3rd generation mobile networks Universal Mobile Telecommunications System (UMTS) and CDMA2000. AKA is based on symmetric keys, and typically runs in a UMTS Subscriber Identity Module (USIM), or a (Removable) User Identity Module ((R) UIM), similar to a smart card. EAP-AKA (Extensible Authentication Protocol Method for UMTS Authentication and Key Agreement) includes optional identity privacy support, optional result indications, and an optional fast reauthentication procedure. The EAP-AKA authentication service is extended to generate a Diameter message Multimedia-Authentication-Request (MAR), with the subscriber identity (IMSI), to the Home Subscriber Server (HSS) when it requires the authentication vectors. The HSS sends a Diameter Multimedia-Authentication-Answer (MAA) back containing the number of quintuplets.

### EAP-AKA-Prime

EAP-AKA-Prime (EAP-AKA') is an EAP authentication method, with a small revision to the existing EAP-AKA method. EAP-AKA' has a new key derivation function, which binds the keys derived within the method to the name of the access network. This limits the effects of compromised access network nodes and keys. EAP-AKA' supports SHA-256 instead of SHA-1.

### EAP-FAST

EAP-FAST is an authentication method which uses the EAP-MSChapV2 method for credential provisioning and EAP-GTC for authentication. Credential provisioning typically occurs only during the client's initial EAP-FAST authentication. Subsequent authentications rely on the provisioned credential and will usually omit the provisioning step.

This authentication protocol is designed to address the performance shortcomings of prior TLS-based EAP methods while retaining features such as identity privacy and support for password-based protocols. The EAP-FAST protocol is described by the IETF draft (draft-cam-winget-eap-fast-00.txt).

### EAP-GTC

This method defined in RFC 2284, is used for transmitting a username and password to an authentication server.



#### Note

It should not be used except as an authentication method for PEAP Version 1 because the password is not protected.

### EAP-LEAP

The new AAA Cisco-proprietary protocol called Light Extensible Authentication Protocol (LEAP) supported by Prime Access Registrar, is a proprietary Cisco authentication protocol designed for use in IEEE 802.11 wireless local area network (WLAN) environments. Important features of LEAP include:

- Mutual authentication between the network infrastructure and the user
- Secure derivation of random, user-specific cryptographic session keys

- Compatibility with existing and widespread network authentication mechanisms (e.g., RADIUS)

**Note**

Prime Access Registrar supports a subset of EAP to support LEAP. This is not a general implementation of EAP for Prime Access Registrar.

The Cisco-Wireless or LEAP is an EAP authentication mechanism where the user password is hashed based on an MD4 algorithm.

**EAP-MD5**

This is another EAP authentication exchange. In EAP-MD5 there is a CHAP-like exchange and the password is hashed by a challenge from both client and server to verify the password. On successful verification, the connection proceeds, although the connection is periodically rechallenge (per RFC 1994).

**EAP-Negotiate**

This is a special service used to select at runtime the EAP service to be used to authenticate the client. It is configured with a list of candidate EAP services that represent the allowable authentication methods in preference order.

EAP-Negotiate is useful when the client population has deployed a mix of different EAP methods that must be simultaneously supported by Prime Access Registrar. EAP-Negotiate solves the problem of distinguishing client requirement by using the method negotiation feature of the EAP protocol.

**EAP-MSChapV2**

EAP-MSChapv2 encapsulates the MSChapV2 protocol (specified by RFC 2759) and can be used either as an independent authentication mechanism or as an inner method for PEAP Version 0 (recommended). This is based on draft-kamath-pppext-eap-mschapv2-00.txt, an informational IETF draft document.

**EAP-SIM**

An access point uses the Prime Access Registrar RADIUS server to perform EAP-SIM authentication of mobile clients. Prime Access Registrar must obtain authentication information from the HLR. Prime Access Registrar contacts the MAP gateway that performs the MAP protocol over SS7 to the HLR, or alternately it can contact the HLR (through STP in some cases) using the SIGTRAN-M3UA interface. The EAP-SIM authentication service is extended to generate a Diameter message Multimedia-Authentication-Request (MAR), with the subscriber identity(IMSI), to the HSS when it requires the authentication vectors. The HSS sends a Diameter Multimedia-Authentication-Answer (MAA) back containing the number of triplets.

**EAP-Transport Level Security (TLS)**

This is an authentication method (described in RFC 2716) which leverages TLS, described in RFC 2246, to achieve certificate-based authentication of the server and the client (optionally). It provides many of the same benefits as PEAP but differs in the lack of support for legacy authentication methods.

**EAP-TTLS**

The Extensible Authentication Protocol Tunneled TLS (EAP-TTLS) is an EAP protocol that extends EAP-TLS. EAP-TTLS extends the authentication negotiation EAP-TLS by using the secure connection established by the TLS handshake to exchange additional information between client and server. It leverages TLS (RFC 2246) to achieve certificate-based authentication of the server (and optionally the client) and creation of a secure session that can then be used to authenticate the client using a legacy mechanism.

EAP-TTLS is a two-phase protocol. Phase 1 conducts a complete TLS session and derives the session keys used in Phase 2 to securely tunnel attributes between the server and the client. The attributes tunneled during Phase 2 can be used to perform additional authentication(s) via a number of different mechanisms.

The authentication mechanisms used during Phase 2 include PAP, CHAP, MS-CHAP, MS-CHAPv2, and EAP. If the mechanism is EAP, then several different EAP methods are possible.

[Table 2-15](#) lists and describes the fields in the EAP Services Details page. The fields listed below are the entire list of all the available types. The fields are displayed based on the type selected.

**Table 2-15 EAP Service Properties**

Fields	Description
Service Name	Required; service name
Incoming Script	Optional script Prime Access Registrar server runs when it receives a request from a client.
Type	Required; must set it to a valid Prime Access Registrar service
Outgoing Script	Optional script Prime Access Registrar server runs before it sends a response to a client
Description (optional)	Optional; description of the PEAP service.
Authentication Timeout	Mandatory; specifies time (in seconds) to wait before an authentication request times out; defaults to 120.
UserService	Required; name of service that can be used to authenticate using cleartext passwords.
ServiceList	List of preconfigured EAP authentication services. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.
Maximum Message Size	Required; indicates the maximum length in bytes that a PEAP message can have before it is fragmented.
Server Certificate File	Required; the full pathname of the file containing the server's certificate or certificate chain used during the TLS exchange. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The two valid encoding prefixes are PEM and DER. If an encoding prefix is not present, the file is assumed to be in PEM format.
Private Key Password	Required; the password used to protect the server's private key.
Server Key File	Required; the full pathname of the file containing the server's RSA private key. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The two valid encoding prefixes are "PEM" and "DER". If an encoding prefix is not present, the file is assumed to be in PEM format.  The following example assumes that the subdirectory <b>pki</b> under <b>/cisco-ar</b> contains the server's certificate file. The file <b>server-key.pem</b> is assumed to be in PEM format. The file extension <b>.pem</b> is not significant.  <b>set ServerRSAKeyFile PEM:/cisco-ar/pki/server-key.pem</b>

Table 2-15 EAP Service Properties (continued)

Fields	Description
CRL Distribution URL	<p>Optional; enter the URL that Prime Access Registrar should use to retrieve the CRL. You can specify a URL that uses HTTP or LDAP.</p> <p>The following is an example for an HTTP URL: &lt;http://crl.verisign.com/pca1.1.1.crl&gt;.</p> <p>The following is an example for an LDAP URL: ldap://209.165.200.225:388/CN=development-CA,CN=acs-west-coast2,CN=CDP,CN=Public Key Services,CN=Services,CN=Configuration,DC=cisco,DC=com</p>
CA Certificate File	<p>Optional; the full pathname of the file containing trusted CA certificates used for client verification. The file can contain more than one certificate, but all certificates must be in PEM format. DER encoding is not allowed.</p>
Certificate Verification Mode	<p>The value is set to optional by default. If set to RequireCertificate, the client certificate will always be verified. If set to optional, client certificate verification happens optionally.</p>
CA Certificate Path	<p>The name of a directory containing trusted CA certificates (in PEM format) used for client verification. This parameter is optional and if it is used there are some special preparations required for the directory it references.</p> <p>Each certificate file in this directory must contain exactly one certificate in PEM format. The server looks up the certificate files using the MD5 hash value of the certificate's subject name as a key. The directory must therefore also contain a set of symbolic links each of which points to an actual certificate file. The name of each symbolic link is the hash of the subject name of the certificate.</p> <p>For example, if a certificate file named <b>ca-cert.pem</b> is located in the CACertificatePath directory, and the MD5 hash of the subject name contained in <b>ca-cert.path.pem</b> is 1b96dd93, then a symbolic link named 1b96dd93 must point to <b>ca-cert.pem</b>.</p> <p>If there are subject name collisions such as multiple certificates with the same subject name, each link name must be indexed with a numeric extension as in 1b96dd93.0 and 1b96dd93.1.</p>
Verification Depth	<p>Optional; specifies the maximum length of the certificate chain used for client verification.</p>
Enable Session Cache	<p>Optional; specifies whether TLS session caching (fast reconnect) is enabled or not. Set to True to enable session caching; otherwise set to False.</p>

Table 2-15 EAP Service Properties (continued)

Fields	Description
Session Timeout	<p>Required; if TLS session caching (fast reconnect) is enabled, SessionTimeout specifies the maximum lifetime of a TLS session. Expired sessions are removed from the cache and will require a subsequent full authentication.</p> <p>SessionTimeout is specified as a string consisting of pairs of numbers and units, where units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks, as in the following:</p> <p><b>Set SessionTimeout “1 Hour 45 Minutes”</b></p>
UseECCCertificate	<p>Determines the applicability of the authentication mechanism in SmartGrid Solutions.</p> <p>When you check this check box, it can use the ECC, RSA, or combination of both certificate for certificate based verification.</p> <p>When you uncheck this check box, it can only use the RSA certificate for certificate based verification. The default location to fetch the certificate file is <b>/cisco-ar/pki</b>.</p>
EnableAutoChaining	<p>When set to TRUE, Prime Access Registrar sends its server certificate chain (Server-Cert -&gt; IntermediateCA -&gt; RootCA) while presenting the server certificate to the client for server side authentication. When set to FALSE, Prime Access Registrar sends only the server certificate (Server-Cert) to the client.</p>
Authentication Service	<p>Specifies the name of the EAP-GTC service used for authentication. The named service must have the UseLabels parameter set to True.</p>
User Prompt	<p>Optional string the client might display to the user; default is Enter password:” Use the <b>set</b> command to change the prompt, as in the following:</p> <p><b>set UserPrompt “Admin Password:”</b></p>
UseLabels	<p>Required; must be set to TRUE for EAP-FAST authentication and set to FALSE for PEAP authentication. Set to FALSE by default.</p>
SystemID	<p>Optional; string that identifies the sender of the MSChapV2 challenge message.</p>
IsWindows7Client	<p>Optional; must be set to TRUE for EAP-MSChapV2 authentication. Set to FALSE by default.</p>
Authority Identifier	<p>Required; a string that uniquely identifies the credential (PAC) issuer. The client uses this value to select the correct PAC to use with a particular server from the set of PACs it might have stored locally.</p>
Authority Information	<p>Required; a string that provides a descriptive text for this credential issuer. The value can be displayed to the client for identification purposes and might contain the enterprise or server names.</p>

**Table 2-15 EAP Service Properties (continued)**

Fields	Description
Credential Life Time	Optional; specifies the maximum lifetime of a Protected Access Credential (PAC). Clients that successfully authenticate with an expired PAC will be reprovisioned with a new PAC.  CredentialLifetime is specified as a string consisting of pairs of numbers and units, where units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks. Credentials that never expire should be specified as Forever.
Provision Service	Required; specifies the name of the EAP-MSChapV2 service used for provisioning.
Provision Mode	Required; specifies the TLS mode used for provisioning. Clients only support the default Anonymous mode.
Always Authenticate	Optional; indicates whether provisioning should always automatically rollover into authentication without relying on a separate session. Most environments, particularly wireless, will perform better when this parameter is set to True, the default value.
SubscriberDBLookup	Specifies the type of communication with the HLR/HSS server.  Based on the type selected, the communication happens with the HLR/HSS server using the diameter Wx interface, MAP protocol, or SIGTRAN-M3UA protocol.  This field is displayed when you select the eap-sim option in the Type field.
Subscriber_DBLookup	Specifies the type of communication with the HLR/HSS server.  Based on the type selected, the communication happens with the HLR/HSS server using the diameter Wx interface, SIGTRAN protocol, or SIGTRAN-M3UA protocol.  This field is displayed when you select the eap-sim, eap-aka, or eap-aka' option in the Type field.
DestinationRealm	Required. Destination realm to send Diameter packets to the remote server. The role of the remote server should be Relay.
PreRequestTranslationScript	Optional. Prime Access Registrar server runs before sending the request to the Diameter remote server. The script can modify the RADIUS packet dictionaries.
PostRequestTranslationScript	Optional. Prime Access Registrar server runs before sending the request to the Diameter remote server. The script can modify the Diameter packet dictionaries.
PreResponseTranslationScript	Optional. Prime Access Registrar server runs after receiving the response from the Diameter remote server. The script can modify the Diameter packet dictionaries.
PostResponseTranslationScript	Optional. Prime Access Registrar server runs after receiving the response from the Diameter remote server. The script can modify the RADIUS packet dictionaries.
FetchAuthorizationInfo	When you check this check box, it fetches MSISDN from HLR.

Table 2-15 EAP Service Properties (continued)

Fields	Description
<b>General tab</b>	
The details in the tab is displayed based on the eap-sim, eap-aka, or eap-aka-prime option you select in the Type field.	
MultipleServersPolicy	<p>Required. Must be set to either Failover or RoundRobin.</p> <p>When set to Failover, Prime Access Registrar directs requests to the first server in the list until it determines the server is offline. At that time, Prime Access Registrar redirects all requests to the next server in the list until it finds a server that is online.</p> <p>When set to RoundRobin, Prime Access Registrar directs each request to the next server in the RemoteServers list to share the resource load across all of the servers listed in the RemoteServers list.</p>
NumberOfTriplets	Required; number of triplets (1, 2, or 3) to use for authentication; default is 2.
PseudonymSecret	<p>Required; the secret string that is used as the basis for protecting identities when identity privacy is enabled. This should be at least 16 characters long and have a value that is impossible for an outsider to guess. The default value is secret. This field is not available if EnableRollingPseudonymSecret field is checked.</p> <p><b>Note</b> It is very important to change PseudonymSecret from its default value to a more secure value when identity privacy is enabled for the first time.</p>
PseudonymRenewtime	<p>Required; specifies the maximum age a pseudonym can have before it is renewed. When the server receives a valid pseudonym that is older than this, it generates a new pseudonym for that subscriber. The value is specified as a string consisting of pairs of numbers and units, where the units might be of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks. The default value is “24 Hours”.</p> <p>Examples are: “8 Hours”, “10 Hours 30 Minutes”, “5 D 6 H 10 M”</p>
PseudonymLifetime	<p>Required; specifies the maximum age a pseudonym can have before it is rejected by the server, forcing the subscriber to authenticate using it’s permanent identity. The value is specified as a string consisting of pairs of numbers and units, where the units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks. It can also be Forever, in which case, pseudonyms do not have a maximum age. The default value is “Forever”.</p> <p>Examples are: “Forever”, “3 Days 12 Hours 15 Minutes”, “52 Weeks”</p>

**Table 2-15 EAP Service Properties (continued)**

Fields	Description
NotificationService	(Optional); Notification service is an authorization service and is used to send a notification code to the client in case of an authorization failure. For more information about the Notification-Code variable, see the “Environment Dictionary” chapter of the <i>Cisco Prime Access Registrar 8.0 Reference Guide</i> . This can be any of the services configured under /radius/services/ except eap services, accounting services, radius-session, radius-query, and diameter.
ReauthenticationTimeout	Required; specifies the time in seconds that reauthentication identities are cached by the server. Subscribers that attempt to reauthenticate using identities that are older than this value will be forced to use full authentication instead. The default value is 3600 (one hour).
EnableReauthentication	Optional; when True, the fast reauthentication option is enabled. The default value is False.
UseOutagePolicyforReauth	Default value is FALSE. When set to TRUE, Prime Access Registrar drops or rejects reauthentication requests as per outage policy when the remote server is down. This can be processed only when there is at least one failed full authentication before proceeding with reauthentication.
OutagePolicy	Required for EAP-SIM, EAP-AKA, and EAP-AKA' services; the default is DropPacket. This property defines how Prime Access Registrar handles requests if all servers listed in the RemoteServers tab are unavailable (that is, all remote RADIUS servers are not available). You must set it to one of the following: <b>AcceptAll</b> , <b>DropPacket</b> , or <b>RejectAll</b> .
UseProtectedResults	Optional; enables or disables the use of protected results messages. Results messages indicate the state of the authentication but are cryptographically protected.
ReauthenticationRealm	Optional; realm to use for reauthentication.
MaximumReauthentications	Required; specifies the maximum number of times a reauthentication identity might be reused before it must be renewed. The default value is 16.
TripletCacheTimeout	Required for eap-sim service; time in seconds an entry remains in the triplet cache. A zero (0) indicates that triplets are not cached. The maximum is 28 days; the default is 0 (no caching).
QuintetCacheTimeout	Required for eap-aka or eap-aka' service; time in seconds an entry remains in the quintet cache. A zero (0) indicates that quintets are not cached. The maximum is 28 days; the default is 0 (no caching).
QuintetGenerationScript	Available for eap-aka or eap-aka' service; script required for quintet generation.
Authentication Timeout	Required; time in seconds to wait for authentication to complete. The default is 2 minutes; range is 10 seconds to 10 minutes.
UseSimDemoTriplets	Optional; set to TRUE to enable the use of demo triplets. This must be disabled for release builds.

**Table 2-15 EAP Service Properties (continued)**

Fields	Description
AlwaysRequestIdentity	Optional; when True, enables the server to obtain the subscriber's identity via EAP/SIM messages instead of relying on the EAP messages alone. This might be useful in cases where intermediate software layers can modify the identity field of the EAP-Response/Identity message. The default value is False.
EnableIdentityPrivacy	Optional; when True, the identity privacy feature is enabled. The default value is False.
Generate3GPPCompliantPseudonym	Optional; the value is set to False by default. If set to TRUE then Prime Access Registrar generates a 12 octet 3GPP compliant pseudonym identity. The Pseudonym username identities are used to protect the privacy of subscriber identities.
SendReAuthIDInAccept	Optional; the value is set to False by default. When set to True, Prime Access Registrar sends SN-Fast-ReAuth-UserName (Starent VSA) in access-accept message.
Outage Script	Optional; if you set this property to the name of a script, Prime Access Registrar runs it when an outage occurs. This property allows you to create a script that notifies you when the RADIUS server detects a failure.
NetworkName	Required for eap-aka-prime service type. Name of the access network for which the authentication is performed. This attribute is captured to ensure that the peer and the server know the name of the access network for performing the EAP authentication.
MapVersion	Required for SIGTRAN-M3UA remote server; select the map version HLR supports.
DiameterInterface	Select SWx or Wx to indicate the Diameter protocol to use for the service.
ProxyService	Select the diameter proxy service to use.
EnableRollingPseudonymSecret	Check this box to activate rolling encryption process that involves generating rolling pseudonym secrets for the service. This option is available only when EnableIdentityPrivacy check box is checked.  For more information about rolling encryption support, see <a href="#">Rolling Encryption Support for Pseudonym Generation in EAP-SIM, EAP-AKA, and EAP-AKA' Services, page 5-61.</a>
EnableEncryptedIMSI	Check this box to look out for encrypted IMSI in the incoming EAP response. For more information, see <a href="#">Support for Decrypting Encrypted-IMSI for EAP-SIM, EAP-AKA, and EAP-AKA' Services, page 5-64.</a>  The following three fields are available when you check this option.
EncryptedIMSIDelimiter	Delimiter value to identify whether the incoming EAP response is encrypted or not. Default value is '\0' (NULL), which indicates the incoming message contains encrypted IMSI.

**Table 2-15 EAP Service Properties (continued)**

Fields	Description
EncryptedIMSIKeyIdDelimiter	<p>Delimiter value to indicate the key identifier from the incoming EAP response. Default value is ',' (comma).</p> <p>The data that exists between the IMSI delimiter ('\0') and Key ID delimiter (',') in the incoming EAP response, is the encrypted IMSI.</p> <p>The data that follows this Key ID delimiter (',') helps the server to locate the private key that can be used to decrypt the incoming encrypted IMSI.</p>
DefaultPrivateKey	Default private key to use for decryption if no private key is configured under <b>Advanced &gt; EncryptedIMSI-PrivateKeys</b> . For more information, see <a href="#">Encrypted IMSI Private Keys</a>
EnableStateStickiness	<p>This field appears for <b>eap-sim</b>, <b>eap-aka</b>, and <b>eap-aka-prime</b> services.</p> <p>Check this box to configure a state attribute value. If this box is unchecked, the Diameter remote server name will be carried as the state attribute value by default.</p>
StateValue	This field appears if <b>EnableStateStickiness</b> is checked. Enter a state attribute value.
<b>Remote Servers tab</b>	
Attribute	Optional; list of remote RADIUS servers which are map gateways. The remote server type must be set to map-gateway. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.

You can use the EAP Services List page for the following:

- [Filtering Records](#)
- [Adding EAP Service Details](#)
- [Editing Records](#)
- [Deleting Records](#)

### Adding EAP Service Details

To add new EAP service details:

- 
- Step 1** Choose **Configuration > Services > EAP**. The EAP Services List page is displayed.
  - Step 2** Click **Add** to add new EAP service details. The EAP Services Details page is displayed.
  - Step 3** Enter the relevant details.
  - Step 4** Click **Submit** to save the specified details in the EAP Services List page. Otherwise click **Cancel** to return to the EAP Services List page without saving the details.

On successful creation of the EAP Service properties, the EAP Services List page is displayed else a respective error message is displayed.

## Diameter Service

Proxy agents assist in routing Diameter messages using the Diameter routing table. Diameter proxy service works in tandem with the rule policy engine to perform the routing for multiple realms or applications. The following are the multiple peer policies supported by the proxy service:

- RoundRobin
- FailOver
- GroupFailOver
- IMSI Range Based.

Table 2-16 lists and describes the fields in the Diameter-Services page. The fields listed below are the entire list of all the available roles. The fields are displayed based on the role selected.

**Table 2-16** *Diameter Service Properties*

Fields	Description
Name	Required; name of the Diameter server.
Description	Optional; description of the Diameter server.
Realm	Required; realm of the route. Must be unique for a route table.
Role	<p>Required; specifies the role that the Diameter entity will play in resolving messages matching the realm.</p> <p>The role can be any one of the following:</p> <p>Relay - Application acting as a Relay Agent.</p> <p>Redirect - Application acting as a Redirect Agent.</p> <p>Proxy - Application acting as a Proxy Agent. When the role is set to Proxy, the IncomingScript and OutgoingScript points are enabled.</p> <p>Local - Application processes the requests locally. When the role is set to Local, the AuthenticationService and AccountingService are enabled.</p> <p>By default, the Proxy option is selected. However, you can select another option from the drop-down list.</p>
Incoming Script	<p>Optional; enabled when role is set to Proxy or Local. When set, must be the name of a known incoming script.</p> <p>Prime Access Registrar runs the IncomingScript before proxying the Diameter packet to the remote Diameter server.</p>
Outgoing Script	<p>Optional; enabled when role is set to Proxy or Local. When set, must be the name of a known outgoing script.</p> <p>Prime Access Registrar runs the OutgoingScript after it receives the response from the remote Diameter server.</p>

**Table 2-16** *Diameter Service Properties (continued)*

<b>Fields</b>	<b>Description</b>
Authentication Service	Required; used when service is configured to process the Diameter requests locally. Set to valid service of type (local/ldap/odbc) to authenticate the user. This field is displayed when you select the role type as 'Local' in the Role field.
AccountingService	Required; used when service is configured to process the accounting requests locally. Set to valid accounting service of type(file/odbc-accounting) to write the accounting records. This field is displayed when you select the role type as 'Local' in the Role field.
Type	Required; specifies the service type.The service type 'Diameter' is automatically displayed in this field.
<b>PEER Statements</b>	
This is displayed when you select the 'Local', 'Relay', or 'Redirect'option in the Role field.	
Name	Required; name of the peer.
Host Name	Required; the hostname or IP address of the peer. The hostname must exist in the client list for the route to be active.
Metric	Required; metric value for the peer entry. The higher the value the lower the preference. The highest value of preference is 0.
VendorSpecific	Required; the default is FALSE. If set to FALSE, the application is ordinary application and user is prompted to enter the ApplicationID. If set to TRUE, the application is a VendorSpecific Application. User is prompted to enter VendorSpecificApplicationID and VendorID.
VendorID	Required; specifies the VendorID for the application. Example: DIAMETER 3GPP Cx APPLICATION VendorSpecificApplicationID 16777216 VendorID 10415
VendorSpecificApplicationID	Required; specifies the integer value for the vendor specific application.
ApplicationID	Required; application used in the route. The application Id should be available in /Advanced/Diameter/Applications.
<b>Applications</b>	
This is displayed when you select the 'Proxy' option in the Role field.	
Name	Required; name of the application.
Description	The description of the application.
ApplicationID	Required; specifies the unique integer value for the application. It represents the application id of the Application used for load balancing the Diameter messages.

**Table 2-16** *Diameter Service Properties (continued)*

Fields	Description
EnableSticky	Required; default is FALSE. If set to True, the sticky entries for load balancing is enabled and the user is prompted to enter the values for StickySessionKey, StickyCreationCmdList, and StickyDeletionCmdList.
MultiplePeersPolicy	Required; Policy used by the Prime Access Registrar server to load balance the peers. Must be set to one of the following: <ul style="list-style-type: none"> <li>RoundRobin—You can list the Diameter remote servers in the tab below.</li> <li>FailOver—You can list the Diameter remote servers in the tab below.</li> <li>GroupFailover—You can create individual groups of Diameter remote servers and list them in the tab below. This option allows you to perform group-based load balancing. For more information, see <a href="#">Group-Based Load Balancing, page 2-24</a>.</li> <li>IMSIRangeBased—You can add the list of IMSI ranges in the tab below.</li> </ul>
PeerTimeoutPolicy/GroupTimeoutPolicy	Required; action to perform when there is a timeout with the Diameter peer or group server.
StickySessionKey	Required; used as the sticky key for mapping the sticky sessions. Set the value to a valid attribute-value pair (AVP) in order to use the sticky key for maintaining Diameter sessions. This ensures that Prime Access Registrar maps the request to the same server for all the subsequent messages using the sticky key. For example, set StickyAVP “Session-Id”.  When the Prime Access Registrar server receives the CCR-I request, Prime Access Registrar extracts the Session-Id from the request packet, maps the Session to the peer configured in the list, and forwards the request to the chosen peer. Prime Access Registrar chooses the same peer for all the subsequent messages(CCR-Update/CCR-Terminate) with same Session-Id.
StickyCreationCmdList	Required; specifies the command list to create the sticky entries. Specify the list of ‘  ’ separated command code, AVP name, and its value to create the sticky sessions.  The following is the StickyCreationCmdList format: <pre>&lt;commandcode1&gt;::&lt;AVPName1=Value1&gt;    &lt;commandcode2&gt;::&lt;AVPName2=Value2&gt;    &lt;commandcode3&gt;</pre> For example, if the sticky session entries need to be created based on command code ‘265’ or based on command code ‘271’ with Accounting-Record-Type value as 2, use the format below: <pre>Set StickyCreationCmdList "265  271:: Accounting-Record-Type=2"</pre>

Table 2-16 Diameter Service Properties (continued)

Fields	Description
StickyDeletionCmdList	<p>Required; specifies the command list to delete the sticky entries. Specify the list of '  ' separated command code, AVP name, and its value to delete the sticky sessions.</p> <p>The following is the StickyDeletionCmdList format:</p> <pre>&lt;commandcode1&gt;::&lt;AVPName1=Value1&gt;    &lt;commandcode2&gt;::&lt;AVPName2=Value2&gt;   &lt;commandcode3&gt;</pre> <p>For example, if the sticky session entries need to be deleted based on command code '271' with Accounting-Record-Type value as 4, use the format below:</p> <pre>Set StickyDeletionCmdList "271:: Accounting-Record-Type=4"</pre>
<b>PEER Definitions Proxy</b>	
Name	Required; name of the peer.
Host Name	Required; hostname or IP address of the peer. The HostName must exist in the client list for the route to be active.
Metric	Required; metric value for this peer entry. The higher the value the lower the preference. The highest value of preference is 0.
Weight	<p>Required; default value is 0. Specifies the weight percentage for which the service needs to load balance the peer.</p> <p><b>Note</b> When you set the weight to a value other than 0, the weight should be in multiples of 10 and the sum of the weights configured in the peer list should be equal to 100.</p>
IMSIRanges	<p>Required; used for load balancing. The value is set to comma separated values of IMSI Ranges.</p> <p>For example, set IMSIRanges "112156000000001-112156001000000,112156010000001-112156011000000"</p> <p><b>Note</b> Prime Access Registrar uses the AVP configured in StickyAVP property to check whether the IMSI is in valid range.</p>
IsActive	Optional; if this is set to true, the new sessions will not go to the peer server. By default, this is set as false.

You can use the Diameter Services List page for the following:

- [Filtering Records](#)
- [Adding Diameter Service Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Diameter Service Details

To add a new Diameter Service details:

- 
- Step 1** Choose **Configuration > Services > Diameter**. The Diameter Services page is displayed.
  - Step 2** Click **Add** to add new Diameter service details. The DIAMETER Services Details page is displayed.
  - Step 3** Specify the required details in the **PEER Statements**, **Applications**, and **PEER Definitions Proxy** specific sections.
  - Step 4** Click **Save DIAMETER Service** to save the specified details in the Diameter Services page. Otherwise click **Cancel** to return to the Diameter Services page without saving the details.

On successful creation of the Diameter Service properties, the Diameter Services page is displayed else a respective error message is displayed.

---



### Note

You may need to enter **PEER Statements**, **Applications**, and **PEER Definitions Proxy** details based on the **Role** that you select in the DIAMETER-Services page.

---

### Adding the PEER Statements Details

To add new PEER Statement details:

- 
- Step 1** Click **Add** to add new PEER Statements details section. The fields specific to PEER Statements are displayed.
  - Step 2** Specify the required details.
  - Step 3** Click **Save** to save the specified details in the PEER Statements section. Otherwise click **Cancel** to return to the PEER Statements section without saving the details.

On successful creation of the Diameter Service properties, the Diameter Services page is displayed else a respective error message is displayed.

---

### Adding the Applications Details

To add new Application details:

- 
- Step 1** Click **Add** to add new Applications details in the Application List section. The fields specific to Applications are displayed.
  - Step 2** Specify the required details.
  - Step 3** Click **Save Appln** to save the specified details in the Application List section. Otherwise click **Cancel Appln** to return to the Application List section without saving the details.
- 

### Adding the PEER Definitions Proxy Details

To add PEER Definitions Proxy details:

- 
- Step 1** Click **Add** to add new Proxy PEER Statements in the PEER Definitions Proxy section. The fields specific to Proxy PEER Statements are displayed.
- Step 2** Specify the required details.
- Step 3** Click **Save** to save the specified details in the Proxy PEER Statements section. Otherwise click **Cancel** to return to the Proxy PEER Statements section without saving the details.
- 

## CommandSets

A command set consists of commands and the action to perform during TACACS+ command authorization.

### Adding a Command Set

To add a new command set:

- 
- Step 1** Choose **Configuration > Command Sets**. Prime Access Registrar lists all the command sets available in the system. You can edit or delete an existing command set.
- Step 2** Click **Add** to add a new command set.
- Step 3** Enter a name and description for the command set.
- Step 4** Provide the Command Set parameters. [Table 2-17](#) lists the parameters in the Add Command section.

**Table 2-17** *Command Set Parameters*

Field	Field Description
Action	Select <b>Permit</b> or <b>Deny</b> to indicate the action to be performed on the command during TACACS+ command authorization.
Command	The command to add in the set. Example: show
Arguments	The arguments for the command. Example: ~/serial*/ <b>Note</b> Prime Access Registrar supports POSIX Extended Regular Expression (ERE) for command arguments.

- Step 5** Click **Add** to add the new command to the set. The command details are displayed in the **Commands** section. You can edit or delete a command from the list as required.
- Step 6** Click **Submit** to save the command set details.
- 

You can use the Command Sets page to perform the following as well:

- [Filtering Records](#)

- [Editing Records](#)
- [Deleting Records](#)

## DeviceAccessRules

A device access rule consists of conditions or expressions and the applicable command sets for TACACS+ command authorization.

### Adding a Device Access Rule

To add a new device access rule:

- 
- Step 1** Choose **Configuration > Device Access Rules**. Prime Access Registrar lists all the device access rules available in the system. You can edit or delete an existing device access rule.
  - Step 2** Click **Add** to add a new device access rule.
  - Step 3** Enter a name and description for the device access rule.
  - Step 4** Choose the default device access action to perform on all commands in the device access rule. Options are **Permit All** or **Deny All**.
  - Step 5** In the Conditions field, include the expressions with **AND** or **OR** conditional operator.
  - Step 6** Select a command set from the drop-down list box and click **Add**. The selected command set is displayed in the Command Set Names list box available. Click **Delete** to remove any command set from the list.
  - Step 7** Provide the expression details for the device access rule. [Table 2-18](#) lists the parameters for adding expressions.

**Table 2-18** Expression Parameters

Field	Field Description
Name	Name of the expression to include in the device access rule.
Description	Description of the expression.
Attribute	Parameter to apply the condition on.
Value	Value of the parameter. <b>Note</b> Prime Access Registrar supports POSIX Extended Regular Expression (ERE) for condition expression value property.

- Step 8** Click **Add** to add the expression to the list-box available in the Condition Expressions section. You can edit or delete the expression from the list as required.
  - Step 9** Click **Submit** to save the device access rule details.
-

## FastRules

FastRules provides a mechanism to easily choose the right authentication, authorization, accounting, and query service(s), drop, reject, or break flows, run a script, choose a session manager and/or a chain of fast rules required for processing a packet.

FastRules has the following capabilities:

- Provides maximum flexibility and ease in matching information in the incoming packets for choosing the appropriate service to apply
- Provides an option to match values in AVPs based on value ranges, exact match, and simple string comparisons using regex
- Provides easy and efficient alternative to rule/policy engine and scripting points for most common use cases—reduces the use of external scripts to choose an appropriate service

For more information about FastRules and the workflow, see [Chapter 11, “Using FastRules to Process Packet Flow.”](#)

### Adding a Fast Rule

To add a new fast rule:

- Step 1** Choose **Configuration > FastRules**. Prime Access Registrar lists fast rules available for RADIUS, Diameter, and TACACS in the respective tabs. You can edit or delete an existing fast rule.
- Step 2** Click **Add** to add a new fast rule. [Table 2-19](#) provides the list of parameters in the FastRules Details page.

**Table 2-19** *FastRules Details*

Field	Field Description
Name	Required; name of the fast rule.
Description	Optional; description of the fast rule.
Protocol	Required; select the type of packet that the fast rule is applicable for from one of the following options: <ul style="list-style-type: none"> <li>• Radius</li> <li>• Diameter</li> <li>• Tacacs</li> </ul>
Condition	Condition based on which the fast rule will be run on the incoming packet. If the condition is success, enter the action to be performed in the Success field. If the condition is failure, enter the action to be performed in the Failure field.
<b>Attributes</b>	
Name	Name of the attribute to include in the condition.
Description	Description of the attribute.
Dictionary	Select type of the dictionary variable as <b>Environment</b> , <b>Request</b> , or <b>Response</b> to map the attribute to.

- Step 3** Add Success and Failure attribute values to the Success Mapping and Failure Mapping fields in the respective sections.
- Step 4** Click **Save** to save the fast rules details.

## Replication

The replication feature of Prime Access Registrar allows you to maintain identical configurations on multiple machines simultaneously. It eliminates the need to have administrators with multiple Prime Access Registrar installations, make the same configuration changes at each of their installations. Instead, only the master's configuration must be changed and the slave is automatically configured eliminating the need to make repetitive, error-prone configuration changes for each individual installation. In addition to enhancing server configuration management, using replication eliminates the need for a hot-standby machine.

Employing Prime Access Registrar's replication feature, both servers can perform RADIUS request processing simultaneously, eliminating wasted resources. It focuses on configuration maintenance only, not session information or installation-specific information.

[Table 2-20](#) lists and describes the fields in the Replication Details page.

**Table 2-20**      *Replication Properties*

Fields	Description
<b>General Properties tab</b>	
Replication Type	Indicates the type of replication
Transaction Sync Interval (in ms)	Duration between periodic transmission of the TransactionSync message expressed in milliseconds. The default is 60000 or 1 minute.
Transaction Archive Limit	The default setting is 100. The value set for RepTransactionArchiveLimit should be the same on the master and the slave.
Replication Secret	The value of this setting must be identical on both the master and the slave.
Is Master	On the master, set RepIsMaster to TRUE. On the slave, set it to FALSE.
Master IP Address	Specifies the IP Address of the master.
Master Port	Specifies the port to be used to send replication messages to the master.
Replication IP Address	The value is set to the IP Address of the machine containing the Prime Access Registrar installation.
Replication Port	Defaults to port1645.
<b>Replication Members tab</b>	
Name	Name of the slave. The name must be unique.

**Table 2-20**     *Replication Properties (continued)*

Fields	Description
IP Address	Indicates the IP Address of the slave.
Port	Port upon which the master will send replication messages to the slave.

You can use the Replication Details page for the following:

- [Filtering Records](#)
- [Adding Replication Details](#)
- [Adding the Replication Member Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Replication Details

To add new replication details:

- 
- Step 1** Choose **Configuration > Replication**. The Replication Details page is displayed.
  - Step 2** Specify the replication details.
  - Step 3** Enter the [Replication Member Details](#), if needed.
  - Step 4** Click **Save** to save the new replication details. Otherwise click **Reset** to restore the default values.

On successful creation of the replication details, a success message is displayed else a respective error message is displayed.

---

## Adding the Replication Member Details

To add new replication member details:

- 
- Step 1** Click the **Replication Members** tab. The List of Replication Members section is displayed.
  - Step 2** Enter the required details.
  - Step 3** Click **Submit** to save the new replication member details.
- 

## RADIUSDictionary

The RADIUS dictionary passes information between a script and the RADIUS server, or between scripts running on a single packet.

Table 2-21 lists and describes the fields in the Add Radius Attributes page. The fields listed below are the entire list of all the available types. The fields are displayed based on the type selected.

**Table 2-21 RADIUS Dictionary Properties**

Fields	Description
Name	Required; must be unique in the RADIUS dictionary list
Description	Optional; description of the attribute
Attribute	Required; must be a number between 1-255. It must be unique within the Attribute dictionary list.
Type	Required; type governs how the value is interpreted and printed.
Minimum	Set to zero
Maximum	Set to 253
Enum Number	Enums allow you to specify the mapping between the value and the strings. After you have established this mapping, Prime Access Registrar then replaces the number with the appropriate string. The min/max properties represent the lowest to highest values of the enumeration.
Enum Equivalent	The value can range from 1 through 255. Click the <b>Add</b> button to save the details and list it in the Enums list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.
Tag	The tag number value can range from 0 through 31. The default value is zero.

You can use the Radius Attributes page for the following:

- [Filtering Records](#)
- [Adding RADIUS Dictionary Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding RADIUS Dictionary Details

To add new RADIUS dictionary details:

- 
- Step 1** Choose **Configuration > Radius Dictionary**. The Radius Attributes page is displayed.
  - Step 2** Click **Add** to add new RADIUS dictionary details. The Add RADIUS Dictionary page is displayed.
  - Step 3** Enter the required details.
  - Step 4** Click **Submit** to save the specified details in the Radius Attributes page. Otherwise click **Cancel** to return to the Radius Attributes page without saving the details.

On successful creation of the Radius Attributes, the Radius Attributes page is displayed else a respective error message is displayed.

## VendorDictionary

The vendor dictionary allows the user to maintain the attributes of the vendor with respect to vendor id, vendor type and the attributes required to support the major NAS.

[Table 2-22](#) lists and describes the fields in the Add Vendor Dictionary page. The fields listed below are the entire list of all the available types. The fields are displayed based on the type selected.

**Table 2-22 Vendor Dictionary Properties**

Fields	Description
Name	Required; must be unique in the Vendor dictionary list
Description	Optional; description of the attribute
Vendor ID	Required; must be a valid number and unique within the entire attribute dictionary
Type	Required; type governs how the value is interpreted and printed.
Minimum	Optional; set to zero
Maximum	Optional; set to 253
Enum Number	Optional; enums allow you to specify the mapping between the value and the strings. After you have established this mapping, Prime Access Registrar then replaces the number with the appropriate string. The min/max properties represent the lowest to highest values of the enumeration.
Enum Equivalent	Optional; the value can range from 1 through 255. Click the <b>Add</b> button to save the details and list it in the Enums list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.
Tag	Optional; the tag number value can range from 0 through 31. The default value is zero.
Vendor Size	Optional; set the vendor size to 8, 16, or 32 bit
HasSubAttributeLengthField	Optional; indicates that the value field of the attribute has the length field for the sub attribute.

You can use the Vendor Dictionary page for the following:

- [Filtering Records](#)
- [Adding Vendor Dictionary Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Vendor Dictionary Details

To add new vendor dictionary details:

- 
- Step 1** Choose **Configuration > Vendor Dictionary**. The Vendor Attributes page is displayed.
  - Step 2** Click **Add** to add new Vendor dictionary details. The Add Vendor Dictionary page is displayed.
  - Step 3** Enter the required details.
  - Step 4** Click **Submit** to save the specified details in the Vendor Attributes page. Otherwise click **Cancel** to return to the Vendor Attributes page without saving the details.

On successful creation of the vendor dictionary details, the Vendor Attributes page is displayed else a respective error message is displayed.



**Note** After adding new vendor dictionary details, you can add vendor attributes details. Or you can also add vendor attributes details by clicking the link in the vendor dictionary list, see [Adding Vendor Attributes](#) for details.

---

## Vendor Attributes

Vendor-specific attributes are included in specific RADIUS packets to communicate prepaid user balance information from the Prime Access Registrar server to the AAA client, and actual usage, either interim or total, between the NAS and the Prime Access Registrar server.

[Table 2-23](#) lists and describes the fields in the Add Vendor Attributes page.

**Table 2-23 Vendor Attribute Properties**

Fields	Description
Name	Required; must be unique in the Vendor attribute list
Description	Optional; description of the attribute
Attribute	Required; must be a valid number and unique within the entire attribute dictionary
Type	Required; type governs how the value is interpreted and printed.
Minimum	Optional; set to zero
Maximum	Optional; set to 253
Enum Number	Optional; enums allow you to specify the mapping between the value and the strings. After you have established this mapping, Prime Access Registrar then replaces the number with the appropriate string. The min/max properties represent the lowest to highest values of the enumeration.

**Table 2-23 Vendor Attribute Properties (continued)**

Fields	Description
Enum Equivalent	Optional; the value can range from 1 through 255. Click the <b>Add</b> button to save the details and list it in the Enums list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.
Tag	Optional; the tag number value can range from 0 through 31. The default value is zero.

You can use the Vendor Attributes page for the following:

- [Filtering Records](#)
- [Adding Vendor Attributes](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Vendor Attributes

To add new Vendor attributes:

- 
- Step 1** Choose **Configuration > Vendor Dictionary**. The Vendor Attributes page is displayed.
  - Step 2** Click the Vendor name link. The Vendor Attributes page is displayed.
  - Step 3** Click **Add** to add new Vendor attributes. The Add Vendor Attributes page is displayed.
  - Step 4** Enter the required details.
  - Step 5** Click **Submit** to save the specified details in the Vendor Attributes page. Otherwise click **Cancel** to return to the Vendor Attributes page without saving the details.

On successful creation of the vendor attributes, the Vendor Attributes page is displayed else a respective error message is displayed.

---

## Vendors

The **Vendor** object provides a central location for specifying all of the request and response processing a particular NAS or Proxy vendor requires. Depending on the vendor, it might be necessary to map attributes in the request from one set to another, or to filter out certain attributes before sending the response to the client. For more information about standard RADIUS attributes, see the “RADIUS Attributes” chapter of the *Cisco Prime Access Registrar 8.0 Reference Guide*.

**Note**

When you have also set **/Radius/IncomingScript**, Cisco Prime Access Registrar runs that script before the vendor's script. Conversely, when you have set a **/Radius/Outgoing** script, Cisco Prime Access Registrar runs the vendor's script before that script.

[Table 2-24](#) lists and describes the fields in the Add Vendor page.

**Table 2-24 Vendor Properties**

Fields	Description
Name	Required; must be unique in the Vendors list.
IncomingScript	Optional; when you specify an IncomingScript, Cisco Prime Access Registrar runs the script on all requests from clients that specify that vendor.
Description	Optional; description of the vendor.
OutgoingScript	Optional; when you specify an OutgoingScript, Cisco Prime Access Registrar runs the script on all responses to the Client.

You can use the Vendors page for the following:

- [Filtering Records](#)
- [Adding Vendor Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Vendor Details

To add new Vendor details:

- 
- Step 1** Choose **Configuration > Vendors**. The Vendors page is displayed.
  - Step 2** Click **Add** to add new Vendor details. The Add Vendor page is displayed.
  - Step 3** Enter the required details.
  - Step 4** Click **Submit** to save the specified details in the Vendors page. Otherwise click **Cancel** to return to the Vendors page without saving the details.

On successful creation of the vendor details, the Vendors page is displayed else a respective error message is displayed.

---

## Translations

**Translations** add new attributes to a packet or change an existing attribute from one value to another. The **Translations** subdirectory lists all definitions of **Translations** the RADIUS server can apply to certain packets.

Under the **/Radius/Translations** directory, any translation to insert, substitute, or translate attributes can be added. The following is a sample configuration under the **/Radius/Translations** directory:

```
cd /Radius/Translations
Add T1
cd T1
Set DeleAttrs Session-Timeout,Called-Station-Id
cd Attributes
Set Calling-Station-Id 18009998888
```

**DeleAttrs** is the set of attributes to be deleted from the packet. Each attribute is comma separated and no spaces are allowed between attributes. All attribute value pairs under the attributes subdirectory are the attributes and values that are going to be added or translated to the packet.

Under the **/Radius/Translations/T1/Attributes** directory, inserted or translated attribute value pairs can be set. These attribute value pairs are either added to the packet or replaced with the new value.

If a translation applies to an Access-Request packet, by referencing the definition of that translation, the Prime Access Registrar server modifies the Request dictionary and inserts, filters, and substitutes the attributes accordingly. You can set many translations for one packet and the Prime Access Registrar server applies these translations sequentially.



### Note

Later translations can overwrite previous translations.

[Table 2-25](#) lists and describes the fields in the Add Translations page.

**Table 2-25** *Translations Properties*

Fields	Description
<b>General Properties tab</b>	
Name	Required; must be unique in the Translations list.
Description	Optional; description of the Translation
Attribute Type	Optional; select either <b>RADIUS</b> or <b>VENDOR</b> . If Vendor is selected, specify the vendor type from the drop-down list. Select the attributes from the available list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.
<b>Attributes tab</b>	
Attribute Type	Optional; select either <b>RADIUS</b> or <b>VENDOR</b> . If Vendor is selected, specify the vendor type from the drop-down list.
Attribute Name	Optional; based on the Attribute Type selected, the attribute name is automated. Set the relevant name for the attribute type selected.
Attribute Value	Optional; set the value for the selected attribute. Click the <b>Add</b> button to save the details and list it in Radius and Value list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.

You can use the Translations page for the following:

- [Filtering Records](#)
- [Adding Translation Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Translation Details

To add new translation details:

- 
- Step 1** Choose **Configuration > Translations**. The Translations page is displayed.
  - Step 2** Click **Add** to add new translations details. The Add Translations page is displayed.
  - Step 3** Enter the required details.
  - Step 4** Click **Add Translation** to save the specified details in the Translations page. Otherwise click **Cancel** to return to the Translations page without saving the details.

On successful creation of the translation details, the Translations page is displayed else a respective error message is displayed.

---

## TranslationGroups

You can add translation groups for different user groups under **TranslationGroups**. All Translations under the Translations subdirectory are applied to those packets that fall into the groups. The groups are integrated with the Prime Access Registrar Rule engine.

The Prime Access Registrar Administrator can use any RADIUS attribute to determine the **Translation Group**. The incoming and outgoing translation group can be different translation groups. For example, you can set one translation group for incoming translations and one for outgoing translations.

Under the **/Radius/TranslationGroups** directory, translations can be grouped and applied to certain sets of packets, which are referred to in a rule. The following is a sample configuration under the **/Radius/TranslationGroups** directory:

```
cd /Radius/TranslationGroups
Add CiscoIncoming
cd CiscoIncoming
cd Translations
Set 1 T1
```

The translation group is referenced through the Prime Access Registrar Policy Engine in the **/Radius/Rules/<RuleName>/Attributes** directory. **Incoming-Translation-Groups** are set to a translation group (for example `CiscoIncoming`) and **Outgoing-Translation-Groups** to another translation group (for example `CiscoOutgoing`).

[Table 2-26](#) lists and describes the fields in the Add Translation Groups page.

**Table 2-26 TranslationGroups Properties**

Fields	Description
Name	Required; must be unique in the Translations list.
Description	Optional; description of the Translation Group.
Translations	Optional; lists of translation. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.

You can use the Translation Groups page for the following:

- [Filtering Records](#)
- [Adding Translation Group Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Translation Group Details

To add new translation group details:

- 
- Step 1** Choose **Configuration > TranslationGroups**. The Translation Groups page is displayed.
  - Step 2** Click **Add** to add new translation group details. The Add TranslationGroup page is displayed.
  - Step 3** Enter the required details.
  - Step 4** Click **Add TranslationGroup** to save the specified details in the Translation Groups page. Otherwise click **Cancel** to return to the Translation Groups page without saving the details.

On successful creation of the translation group details, the Translation Groups page is displayed else a respective error message is displayed.

---

## Diameter

Diameter is a computer networking protocol for Authentication, Authorization and Accounting (AAA). It is a successor to RADIUS or an enhanced version of the RADIUS protocol. It includes numerous enhancements in all aspects, such as error handling and message delivery reliability. It extracts the essence of the AAA protocol from RADIUS and defines a set of messages that are general enough to be the core of the Diameter Base protocol. The various applications that require AAA functions can define their own extensions on top of the Diameter base protocol, and can benefit from the general capabilities provided by the Diameter base protocol.

The following sections can be used to configure Diameter transport management properties, session management properties, add new application, commands associated with it and application specific AVPs:

- [General](#)

- [Session Management](#)
- [Applications](#)
- [Commands](#)
- [DiameterAttributes](#)

## General

This section explains how to set Diameter general configuration such as product name, version, and transport management properties.

### Setting General Diameter Parameters

[Table 2-27](#) lists and describes the fields in the General Diameter Properties page.

**Table 2-27** General Diameter Properties

Fields	Description
<b>General section</b>	
Product	Optional; name of the product.
AuthApplicationIdList	Specifies the list of AuthApplications that the Prime Access Registrar server registers to Diameter Base stack during start up. It is a combination of Auth ApplicationId's separated by a colon.
Version	Optional; version number.
AcctApplicationIdList	Specifies the list of AcctApplications that the Prime Access Registrar server registers to Diameter Base stack during start up. It is a combination of Acct ApplicationId's separated by a colon.
<b>Transport Management section</b>	
Identity	Required; identity of the system on which Diameter application is running. Must be set to a valid resolvable string.
Realm	Required; must be set to a valid Realm in the domain.
EnableIPV6	Required; if set to TRUE it enables IPV6 for the Diameter application.
ValidateIncomingMessages	Check the box to validate incoming messages.
ValidateOutgoingMessages	Check the box to validate outgoing messages.
MaximumNumberofDiameterPackets	Required; the maximum number of Diameter packets that can be processed.
DiameterPacketSize	Required; the Diameter packet size that can be processed. An incoming Diameter packet with a packet size more than the value set in this field will be dropped.
WatchdogTimeout	Required; specifies the time interval between watch dog messages.
ReserveDiameterPacketPool	Percentage of the Diameter packet pool to reserve for the Diameter remote server responses.

**Table 2-27** General Diameter Properties (continued)

Fields	Description
TCPListenPort	Required; port number on which the Prime Access Registrar server listens for TCP peer connections.
SCTPListenPort	Required; port number on which the Prime Access Registrar server listens for SCTP peer connections.
ReconnectInterval	Required; specifies the time interval between which Prime Access Registrar server attempts to connect to a disconnected peer. If set to 0, then no attempt will be made to connect to a disconnected peer.
MaxReconnections	Required; specifies the number of times Prime Access Registrar server tries to make a reconnection attempt. If set to 0, then no attempt will be made to reconnect.
RequestRetransmissionInterval	Required; the time for which retransmission of pending requests will be done. If set to 0, then no attempt will be made to retransmit.
MaxRequestRetransmissionCount	Required, maximum number of times Prime Access Registrar server tries to retransmit a pending request. If set to 0, then no attempt will be made to retransmit.
Receive BufferSize	Required; initial size of buffer that is preallocated for message reception.
<b>SCTPOptions Section</b>	
MaxInitRetry	Maximum number of retries for INIT message to open a connection. Valid range is 0 - 255. Set to 0 to retry indefinitely.
MaxInboundStream	Maximum number of incoming streams per connection. Valid range is 1 - 65545.
MaxOutboundstream	Maximum number of outgoing streams per connection. Valid range is 1 - 65545.
HeartbeatInterval	Default heartbeat interval for a connection.
EnableHeartbeat	Indicates whether to enable or disable heartbeat to monitor the connections and allow earlier detection of loss connections.
AdvertisedHostName	Optional, specifies the local hostname address that will be advertised by the Prime Access Registrar server to other peers during CER/CEA exchange.  For example: AdvertisedHostNames = toby-ar1.cisco.com

### Setting Up the General Diameter Parameters

To set up the general Diameter parameters:

**Step 1** Choose **Configuration > Diameter > General**. The General Diameter page is displayed.

**Step 2** Specify the required details.

**Step 3** Click **Set** to save the specified details.

On successful creation of the general Diameter parameters, a success message is displayed else a respective error message is displayed.

## Session Management

Diameter Base protocol stack provides the functionality of Session Management. Base Stack maintains sessions separately for authentication and accounting messages. Session-Id AVP is used to identify the user session.

Table 2-28 lists and describes the fields in the Session Management page.

**Table 2-28** Session Management Properties

Fields	Description
<b>Session Management section</b>	
MaxNumberOfSessions	Required; specifies the maximum number of concurrent Diameter sessions the Prime Access Registrar server will maintain. These sessions include both Auth and Acct sessions.
<b>AuthSessions section</b>	
EnableStatefulSessions	If set to TRUE, the server will enforce stateful sessions and the client will hint for stateful sessions. Default Value is TRUE. Set the property to FALSE to disable stateful sessions.
AuthSessionTimeout	Required; specifies the timeout in seconds before a session requires reauthentication.
LifeTimeTimeout	Required; specifies the timeout in seconds before a session is terminated regardless of whether the session has been re-authenticated.
GracePeriodTimeout	Required; specifies the grace period after the life timeout and before the full termination of the session.
AbortRetryTimeout	Required; specifies the timeout between the subsequent Abort Session Request (ASR) messages if the initial attempt fails.
<b>AcctSessions section</b>	
AcctSessionTimeout	Required; specifies the timeout in seconds before a session requires reauthentication.
InterimInterval	Required; specifies the interim interval dictated to the client if the entity is a server or hint to the server if the entity is a client.
RealTime	Required; RealTime value dictated to the client.

## Setting Session Management Properties

To set up the session management properties:

- Step 1** Choose **Configuration > Diameter>SessionManagement**. The Session Management page is displayed.
- Step 2** Enter the required details and click **Set**.
- On successful creation of the parameters, a success message is displayed else a respective error message is displayed.

## Applications

A Diameter application is not a software application, but a protocol based on the Diameter base protocol (defined in RFC 6733). Each application is defined by an application identifier and can add new command codes and/or new mandatory AVPs.

When you click the Add button in the Applications page, the Application Details page is displayed. [Table 2-29](#) lists and describes the fields in the Application Details page.

**Table 2-29** *Diameter Application Properties*

Fields	Description
Name	Required; name of the application.
Description	Optional; description of the application.
VendorSpecific	Required; the default is FALSE. If set to FALSE, the application is ordinary application and user is prompted to enter the ApplicationID. If set to TRUE, the application is a VendorSpecific Application. User is prompted to enter VendorSpecificApplicationID and VendorID.
AuthApplication	Required; if set to TRUE the application represents AuthApplication else it represents Accounting Application.
ApplicationURI	Optional; specifies the URI of the Application. Eg: "ftp://ftp.ietf.org/internet-drafts/draft-ietf-aaa-diameter-nasreq-12.txt"
ApplicationID	Required; specifies the unique integer value for the application. The following are examples of Diameter application: NASREQ 1 Mobile-IP 2 Diameter Base Accounting 3 <b>Note</b> ApplicationURI property must be set to 0 for Base Protocol.
VendorSpecificApplicationID	Required; specifies the integer value for the vendor specific application.

**Table 2-29** *Diameter Application Properties (continued)*

Fields	Description
VendorID	Required; specifies the VendorID for the application. Example: DIAMETER 3GPP Cx APPLICATION VendorSpecificApplicationID 16777216 VendorID 10415
Commands	Required; an indexed list from 1 to <n>. Each entry in the list is the name of the command. It specifies the list of commands associated with the application.  To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.

You can use the Applications page for the following:

- [Filtering Records](#)
- [Adding Diameter Application Details](#)
- [Commands](#)
- [Editing Records](#)
- [Deleting Records](#)

### Adding Diameter Application Details

To add new Diameter application details:

- 
- Step 1** Choose **Configuration > Diameter > Applications**. The Applications page is displayed.
  - Step 2** Click **Add**. The Application Details page is displayed.
  - Step 3** Enter the relevant details.
  - Step 4** Click **Add Application** to save the specified details in the Application Details page. Otherwise click **Cancel** to return to the Applications page without saving the details.

On successful creation of the Applications details, a success message is displayed else a respective error message is displayed.

---

### Commands

Each command in Diameter is associated with a command code. The command can be a request command or an answer command which is identified by the 'R' bit in the Command Flags field of the Diameter header.

When you click the Add button in the commands page, the Command Details page is displayed.

[Table 2-30](#) lists and describes the fields in the Command Details page.

**Table 2-30** *Diameter Commands Properties*

Fields	Description
Name	Required; name of the command.
Description	Optional; description of the command.
Command Code	Required; specifies the integer code of the command.
EnableProxyBit	Required; default is TRUE. When enabled it represents the message is proxiabile.
RequestFixed tab	Defines the fixed position of AVP in a request message.
RequestRequired tab	The AVP must be present and can appear anywhere in the request message.
RequestOptional tab	The AVP name in optional cannot evaluate to any avp name which is included in a fixed or required directory. The avp can appear anywhere in the request message.
AnswerFixed tab	Defines the fixed position of AVP in the answer message.
AnswerRequired tab	The AVP must present and can appear anywhere in the answer message.
AnswerOptional tab	The AVP name in optional cannot evaluate to any avp name which is included in a fixed or required directory. The avp can appear anywhere in the answer message.

You can click the Add button in the Command Details page to add the AVP details. [Table 2-31](#) lists and describes the fields displayed on clicking the **Add** button.

**Table 2-31** *Request/Answer Msg AVP Properties*

Fields	Description
Name	Required; name of the AVP.
Description	Optional; description of the AVP.
Min	Specifies the minimum number of times AVP element may be present in a request. The default value is 0.
Max	Specifies the maximum number of times the element may present in a request. A value of zero implies AVP is not present in the request.

## Adding Diameter Commands

To add the Diameter commands:

- Step 1** Choose **Configuration > Diameter > Commands**. The Commands page is displayed.
- Step 2** Click **Add**. The Add Commands page is displayed.
- Step 3** Enter the relevant details.
- Step 4** Click the required tab and click **Add** to enter the AVP details.

- Step 5** Click **Save** to save the AVP details or click **Cancel** to exit the page without saving the details.
- Step 6** Click **Add Command** to save the specified details in the Add Commands page. Otherwise click **Cancel** to return to the Commands page without saving the details.

The Commands page is displayed with the newly added details or a respective error message is displayed.

## DiameterAttributes

You can define the attributes to use in the Diameter EAP application.

[Table 2-32](#) lists and describes the fields in the DiameterAttributes page.

**Table 2-32** *Diameter Attributes Properties*

Fields	Description
Name	Required; name of the attribute.
Description	Optional; description of the attribute.
Attribute	Required; attribute value.
VendorID	Required; Vendor ID of the Diameter application.
Mandatory	Indicates whether the attribute is mandatory or not. Options are May, Must, and MustNot.
May-Encrypt	Choose <b>Yes</b> or <b>No</b> to indicate whether the attribute value can be encrypted or not.
Protected	Indicates whether the attribute value is protected or not. Options are May, Must, and MustNot.
Type	Choose the type of the attribute.
Minimum	Minimum value for the attribute.
Maximum	Maximum value for the attribute.

## Adding Diameter Attributes

To add the Diameter attributes:

- Step 1** Choose **Configuration > Diameter > DiameterAttributes**. The DiameterAttributes page is displayed.
- Step 2** Click **Add**.
- Step 3** Provide the relevant details as explained in [Table 2-32](#).
- Step 4** Click **Add DiameterAttributes** to save the specified details. Otherwise click **Cancel** to return to the previous page without saving the details.

The DiameterAttributes page is displayed with the newly added details or a respective error message is displayed.

## Advanced

Advanced objects allow configuring system-level properties and the Attribute dictionary. Under normal system operation, the system-level properties should not be changed.

The following list helps you in defining the system-level properties and attribute dictionary:

- [Default](#)
- [BackingStore/ServerParam](#)
- [RemoteSessionServer](#)
- [SNMP and Server Monitor](#)
- [DDNS](#)
- [Encrypted IMSI Private Keys](#)
- [ODBC DataSources](#)
- [Log](#)
- [Ports](#)
- [Interfaces](#)
- [Attribute Groups](#)

## Default

This feature of GUI allows you in configuring the default values for other functionalists of GUI. The configurations set in this feature reflects on all the other features.

[Table 2-33](#) lists and describes the fields in the Default Advanced Details page.

**Table 2-33**      *Default Configuration Details*

Fields	Description
<b>Default section</b>	
AAAFServiceSyncInterval	Required; specified in milliseconds, the default is 75. This property governs how often the file AAA service processes accounting requests and writes the accounting records to the file. You can lower the number to reduce the delay in acknowledging the <b>Account-Request</b> at the expense of more frequent flushing of the accounting file to disk. You can raise the number to reduce the cost of flushing to disk, at the expense of increasing the delays in acknowledging the <b>Accounting-Requests</b> . The default value was determined to provide a reasonable compromise between the two alternatives.
RemoteRadiusServerInterface	When set, specifies the local interface to bind to when creating the RemoteRadiusServer socket. If not set, the Prime Access Registrar binds to IPADDR_ANY.

**Table 2-33** *Default Configuration Details (continued)*

Fields	Description
MaximumNumberOfXML-Packets	Required when using identity caching. Indicates the maximum number of XML packets to be sent or received. The minimum value is 1 and the maximum is a 32-bit unsigned integer. The default is 1024.
MaximumODBCResultSize	Required; specifies maximum size in bytes for an ODBC mapping. This parameter affects both ODBC result sizes and the trace log buffer for tracing script calls that access any of the dictionaries. (Default value is 256.)
XMLUDPPacketSize	Required when using identity caching. Indicates the maximum size of XML packets to be sent or received. The minimum value is 1 and the maximum is a 32-bit unsigned integer. The default is 4096.
InitialBackgroundTimerSleepTime	Required; the default is 5. This property specifies the amount of time the time queue should initially sleep before beginning processing. This property is only used for initial synchronization and should not be changed.
RemoteLDAPServerThreadTimerInterval	Required; specified in milliseconds, the default is 10. This property governs how often the ldap RemoteServer thread checks to see if any results have arrived from the remote LDAP server. You can modify it to improve the throughput of the server when it proxies requests to a remote LDAP server.
AdvancedDuplicateDetectionMemoryInterval	Required when the Advanced Duplicate Detection feature is enabled. This property specifies how long (in milliseconds) Cisco Prime Access Registrar should remember a request. You must specify a number greater than zero. The default is 10,000.
RollingEncryptionKeyChangePeriod	Used in conjunction with the session-cache ResourceManager, this property specifies the length of time a given EncryptionKey will be used before a new one is created. When the session-cache ResourceManager caches User-Password attributes, Prime Access Registrar encrypts the User-Password so it is not stored in memory or persisted on disk in clear text. Prime Access Registrar uses up to 255 encryption keys, using a new one after each RollingEncryptionKeyChangePeriod expires. If RollingEncryptionKeyChangePeriod is set to 2 days, Prime Access Registrar will create and begin using a new EncryptionKey every two days. The oldest key will be retired, and Prime Access Registrar will re-encrypt any User-Passwords that used the old key with the new key. This way, if the RollingEncryptionKeyChangePeriod is set to 1 day, no key will be older than 255 days.
DefaultReturnedSubnetSizeIfNoMatch	Optional; used with the ODAP feature and reflects the returned size of the subnet if no matched subnet is found. There are three options to select if an exactly matched subnet does not exist: Bigger, Smaller, and Exact. The default is Bigger.
ODBCEnvironmentMultiValueDelimiter	Optional; allows you to specify a character that separates multivalued attributes in the marker list when using ODBC accounting

**Table 2-33** *Default Configuration Details (continued)*

Fields	Description
RemoteSigtranServerThread-TimerInterval	Required; specified in milliseconds, the default is 10. This property governs how often the sigtran RemoteServer thread checks to see if any results have arrived from the remote HLR/AuC server. You can modify it to improve the throughput of the server when it proxies requests to a remote sigtran server.
AdditionalNativeOracleErrors	Optional; 5 digit Oracle native error in order to disconnect the ODBC/OCI remote servers.
EnableLengthFlag	Check this box to enable the length flag.
FlushDiskInBackground	Check this box to allow Prime Access Registrar to flush the accounting record to disk before it acknowledges the request packets.
SendOpCodeInISDResponse	Check this box to send operator code in the ISD response.
EnableRoutingContextInM3UA	Check this box to enable routing context in M3UA.
DefaultRadiusSharedSecret	Enter the default shared secret for RADIUS server.
ReserveRADIUSPacketPool	Percentage of the RADIUS packet pool to reserve for the RADIUS remote server responses.
EnableLocationCapability	<p>Check the box to enable location-based attributes within RADIUS and Diameter that can be used to convey location-related information for authentication and accounting exchanges.</p> <p>If this option is enabled, Prime Access Registrar retrieves the location information from the client and processes the incoming packet for AA services.</p> <p>For more information on location information delivery flows, refer to RFC 5580. For information on location-based attributes in Prime Access Registrar, see the “Environment Dictionary” chapter of the <i>Cisco Prime Access Registrar 8.0 Reference Guide</i>.</p>
TLSv1Enabled	Applicable only for Diameter; Set to TRUE to use TLS version 1.0 and above for Diameter connection. Set to FALSE to use TLS version greater than 1.0 for Diameter connection.
DiameterSessionRestorationPurgeTime	<p>The time at which Prime Access Registrar must run the Diameter session restoration process. Format is HH:MM:SS (24 hrs format) and default value is 02:00:00.</p> <p>Recommended time is when the incoming traffic is minimal.</p> <p><b>Note</b> This time should always be two hours behind the Diameter stale session purge time.</p>
DiameterStaleSessionPurgeTime	<p>The time at which Prime Access Registrar must check for Diameter stale sessions. Format is HH:MM:SS (24 hrs format) and default value is 00:00:00.</p> <p>Recommended time is when the incoming traffic is minimal.</p>
SocketWaitTime	Fixed wait time for receiving socket data.
ServerStatusSharedSecret	The shared secret for the RADIUS remote server status.
UserLogDelimiter	Delimiter value to be used for the user/subscriber log data.

**Table 2-33** *Default Configuration Details (continued)*

<b>Fields</b>	<b>Description</b>
DiameterStaleConnectionDeletionTimeout	The timeout value in milliseconds, after which Prime Access Registrar deletes the Diameter stale peer connections. Default value is 300000.

Table 2-33 Default Configuration Details (continued)

Fields	Description
<b>AR Flags section</b>	
HideSharedSecretAndPrivateKeys	<p>Optional; the default value is TRUE.</p> <p>The HideSharedSecretAndPrivateKeys property hides:</p> <ul style="list-style-type: none"> <li>• The secret that is shared between a RADIUS Client and a RADIUS Server or between two RADIUS servers in a RADIUS proxy scenario.</li> <li>• The PrivateKeyPassword under the certificate-based EAP services.</li> </ul> <p>When this property is set to TRUE, the following properties are displayed as &lt;encrypted&gt;:</p> <ul style="list-style-type: none"> <li>• PrivateKeyPasswords in: <ul style="list-style-type: none"> <li>– peap-v0 service</li> <li>– peap-v1 service</li> <li>– eap-tls service</li> <li>– eap-ttls service</li> <li>– eap-fast service</li> </ul> </li> <li>• SharedSecret in: <ul style="list-style-type: none"> <li>– RemoteServers of type RADIUS</li> <li>– RemoteServers of type map-gateway</li> <li>– Clients object</li> <li>– Resource Manager of type usr-vpn under Gateway subobject</li> </ul> </li> <li>• PseudonymSecret in eap-sim service</li> <li>• DynamicAuthSecret under DynamicAuthorizationServer subject in Clients object</li> <li>• RepSecret under Replication</li> <li>• Secret in /radius/advanced/DDNS/TSIGKeys</li> </ul> <p>When the value for this property is set to FALSE, all the above properties are displayed in clear text.</p>
ListenForDynamicAuthorizationRequests	<p>Must be set to TRUE when using the Change of Authorization (CoA) feature or Packet of Disconnect (POD) feature. Default is FALSE.</p>
RequireNASsBehindProxy-BeInClientList	<p>Optional; the default is FALSE. If you accept the default, Cisco Prime Access Registrar only uses the source IP address to identify the immediate client that sent the request. Leaving it FALSE is useful when this RADIUS Server should only know about the proxy server and should treat requests as if they came from the proxy server. This might be the case with some environments that buy bulk dial service from a third party and thus do not need to, or are unable to, list all of the NASs behind the third party's proxy server. When you set it to TRUE, you must list all of the NASs behind the Proxy in the Clients list.</p>

Table 2-33 Default Configuration Details (continued)

Fields	Description
UseAdvancedDuplicateDetection	Required; the default is FALSE. Set this property to TRUE when you want Cisco Prime Access Registrar to use a more robust duplicate request filtering algorithm.
DetectOutOfOrderAccountingPackets	<p>Optional; used to detect accounting packets that arrive out of sequential order. The default is FALSE. This property is useful when using accounting and session management in a RADIUS proxy service.</p> <p>When the DetectOutOfOrderAccountingPacket property is enabled (set to TRUE), a new <i>Class</i> attribute is included in all outgoing Accept packets. The value for this Class attribute will contain the session magic number. The client will echo this value in the accounting packets, and this will be used for comparison.</p> <p>The session magic number is a unique number created for all sessions when the session is created or reused and the DetectOutOfOrderAccountingPacket property is set to TRUE. The DetectOutOfOrderAccountingPacket property is used to detect out-of-order Accounting-Stop packets in roaming scenarios by comparing the session magic number value in the session with the session magic number value contained in the Accounting packet.</p> <p>The value of 0xffffffff is considered by the Prime Access Registrar server to be a wild card magic number. If any accounting stop packets contain the value of 0xffffffff, it will pass the session magic validation even if the session's magic number is something else.</p> <p>The format of the class attribute is as follows:</p> <p style="padding-left: 40px;">&lt;4-byte Magic Prefix&gt;&lt;4-byte server IP address&gt;&lt;4-byte Magic value&gt;</p>
<b>Java and EAP Parameters section</b>	
ClasspathForJavaExtensions	<p>A string which is the classpath to be used to locate Java classes and jar files containing the classes required for loading the Java extensions, either Java extension points or services.</p> <p><b>Note</b> The classpath will always contain the directory <b>\$INSTALL-DIR/scripts/radius/java</b> and all of the jar files in that directory.</p>
JavaVMOptions	A string that can contain options to be passed to the JRE upon startup. JavaVMOptions should be used only when requested by Cisco TAC.
EapBadMessagePolicy	<p>Set to one of two values: SilentDiscard (the default) or RejectFailure.</p> <p>When set to SilentDiscard, the Prime Access Registrar server silently discards and ignores bad EAP messages unless the protocol specification explicitly requires a failure message.</p> <p>When set to RejectFailure, the Prime Access Registrar server sends RADIUS Access-Rejects messages with embedded EAP-Failure in response to bad EAP messages as described in Internet RFC 3579.</p>

**Table 2-33** *Default Configuration Details (continued)*

Fields	Description
CertificateDBPath	Required if you are using an LDAP RemoteServer and you want Prime Access Registrar to use SSL when communicating with that LDAP RemoteServer. This property specifies the path to the directory containing the client certificates to be used when establishing an SSL connection to an LDAP RemoteServer. This directory must contain the <b>cert7.db</b> and <b>cert5.db</b> certificates and the <b>key3.db</b> and <b>key.db</b> files database used by Netscape Navigator 3.x (and above) or the <b>ServerCert.db</b> certificate database used by Netscape 2.x servers.
UISessionTimeoutInMins	Required; maximum value is 30 minutes.  When set to a non-zero value, an administrator will be able to hold only one active session. This includes GUI, CLI, and REST API sessions.  GUI or CLI session will be logged out automatically, if left inactive for the configured timeout value.  After three consecutive failed login attempts, administrator will be blocked for the configured time i.e. the administrator will be able to login only after the time (in mins) mentioned in this field.

### Setting Default Configuration

To set up the default configuration details:

- 
- Step 1** Choose **Configuration > Advanced > Default**. The Default Advanced Details page is displayed.
  - Step 2** Enter the relevant details.
  - Step 3** Click **Set** to save the specified details in the Default Advanced Details page. Otherwise, click **Reset** to restore the default values. On successful creation of the default configurations, a success message is displayed else a respective error message is displayed.
- 

### BackingStore/ServerParam

The Backing Store is a Parsing Tool which helps you in analyzing the session backing store files. It retrieves the information on RADIUS sessions, clears phantom sessions details manually and processes the binary log files information to user-readable format.

The Server parameters are set to configure objects to remote server using the relevant aregcmd commands.

Table 2-34 lists and describes the fields in the Backing/ServerParam Advanced Details page.

**Table 2-34 BackingStore/ServerParameter Properties**

Fields	Description
<b>Backing Store section</b>	
SessionBackingStoreSyncInterval	Sessions will be written to the backing store at this interval
PacketBackingStoreSyncInterval	The minimum value is 1 and the maximum is a 32-bit unsigned integer. The default is 75.
SessionBackingStorePruneInterval	<p>Required; specifies the sleep time interval of the session backing store pruning thread. The recommended and default value is 6 hours, but you can modify this based on the traffic patterns you experience.</p> <p>With SessionBackingStorePruneInterval set to 6 hours, pruning will occur 6 hours after you restart or reload the Prime Access Registrar server and recur every 6 hours.</p> <p>You can set a very low value for this property to make pruning continuous, but there might not be enough data accumulated for the pruning to occur and pruning might be less effective compared to the default setting.</p>
PacketBackingStorePruneInterval	<p>Required; specifies the sleep time interval of the packet backing store pruning thread. The recommended value is 6 hours, but you can modify this based on the traffic patterns you experience.</p> <p>When PacketBackingStorePruneInterval is set to 6 hours, pruning will occur 6 hours after you restart or reload the Prime Access Registrar server and recur every 6 hours.</p> <p>You can set a very low value for this property to make pruning continuous, but there might not be enough data accumulated for the pruning to occur and pruning might be less effective compared to the default setting.</p>
BackingStoreDiscThreshold	<p>Required; the default is 10 gigabytes. The value of BackingStoreDiscThreshold is made up of a number of units which can be K, kilobyte, or kilobytes, M, megabyte, or megabytes, or G, gigabyte, or gigabytes.</p> <p>BackingStoreDiscThreshold is used with session management and ODBC accounting and ensures that any data log files generated will not cross the BackingStoreDiscThreshold.</p>

**Table 2-34** *BackingStore/ServerParameter Properties (continued)*

Fields	Description
SessionPurgeInterval	<p>Optional; the SessionPurgeInterval property determines the time interval at which to check for timed-out sessions. If no value is set, the session timeout feature is disabled. The checks are performed in the background when system resources are available, so checks might not always occur at the exact time set.</p> <p>The minimum recommended value for SessionPurgeInterval is 60 minutes. The SessionPurgeInterval value is comprised of a number and a units indicator, as in n units, where a unit is one of minutes, hours, days, or weeks.</p>
StaleSessionTimeout	<p>Required; the default value is “1 hour.” Specifies the time interval to maintain a session when a client does not respond to Accounting-Stop notification.</p> <p>When the Prime Access Registrar server does not receive an Accounting-Response from a client after sending an Accounting-Stop packet, Prime Access Registrar maintains the session for the time interval configured in this property before releasing the session.</p> <p>This property is stored as a string composed of two parts: a number and a unit indicator (&lt;n&gt; &lt;units&gt;) similar to the MaxFileAge property where the unit is one of: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, or Weeks.</p>
NumberOfRadiusIdentifiersPerSocket	<p>This represents the number of RADIUS Identifiers that Prime Access Registrar can use per source port, while proxying requests to remote servers.</p> <p>To use a different source port for every request that is proxied, you need to set the value of this property to one.</p>
EnableStickySessionCount	<p>Required; either True or False and the default value is True. When set to True, Prime Access Registrar displays the peer specific stats showing the number of sticky sessions associated with a peer for Diameter proxy service in name_radius_log file.</p>
StickySessionCountInterval	<p>Required; specified in milliseconds and the default is 60000. When the EnableStickySessionCount is set to True, this field specifies how often the Diameter proxy service will display the number of sticky sessions associated with a peer.</p>
StickySessionSyncInterval	<p>Required; specified in milliseconds and the default value is 500. Specifies how often the Diameter proxy service will write the sticky sessions to a file located in /cis-co-ar/temp/__sticky_sessions_store location.</p>

Table 2-34 BackingStore/ServerParameter Properties (continued)

Fields	Description
<b>Server Parameters section</b>	
MaximumNumberOfRadiusPackets	Required; the default is 8192. This is a critical property you should set high enough to allow for the maximum number of simultaneous requests. When more requests come in than there are packets allocated, Cisco Prime Access Registrar will drop those additional requests.
NumberOfRemoteUDPServerSocket	<p>Required; the default value for this property is 4.</p> <p>The NumberOfRemoteUDPServerSockets property allows you to configure the number of source ports used while proxying requests to a remote RADIUS server. If the NumberOfRemoteUDPServerSockets property is set to a value <i>n</i>, all remote servers share and use <i>n</i> sockets.</p> <p>The NumberOfRemoteUDPServerSockets value comprises a number, as in <i>n</i>, where <i>n</i> should be less than or equal to the current process file descriptor limit divided by 4.</p> <p><b>Note</b> By default, the RADIUS process supports up to 1024 file descriptors. To increase the file descriptors, stop the arserver; in the arserver script, specify the required value to "NUMBER_OF_FILE_DESCRIPTOR" and restart the server. The value for "NUMBER_OF_FILE_DESCRIPTOR" should be in the range between 1024 to 65535.</p>
MemoryLimitForRadiusProcess	This property is used to avoid crashing of the RADIUS process. The default value is 3500 Megabytes. This property is under <b>/radius/advanced</b> . When the RADIUS process uses memory more than the configured limit, further sessions are not created and Prime Access Registrar rejects further incoming requests.
MemorySizeCheckInterval	This property is used to avoid crashing of the RADIUS process. This is used in conjunction with <b>MemoryLimitForRadiusProcess</b> . The default value is 5 minutes. <b>MemorySizeCheckInterval</b> is a hidden parameter in mcd database. To modify the default value, you need to export the mcd database. Typically, a separate thread is created to monitor the RADIUS process memory usage for every 5 minutes.
UDPPacketSize	Required; the default is 4096. RFC 2138 specifies the maximum packet length can be 4096 bytes. Do not change this value.

**Table 2-34** *BackingStore/ServerParameter Properties (continued)*

Fields	Description
PerPacketHeapSize	Required; the default is 6500. This property sets the size of the initial heap for each packet. The heap is the dynamic memory a request can use during its lifetime. By preallocating the heap size at the beginning of request processing, we can minimize the cost of memory allocations. If PerPacketHeapSize is too low, Prime Access Registrar will ask the system for memory more often. If PerPacketHeapSize is too high, Prime Access Registrar will allocate too much memory for the request causing the system to use more memory than required.
MinimumSocketBufferSize	Required; the default is 65536 (64 K). This property governs how deep the system's buffer size is for queueing UDP datagrams until Cisco Prime Access Registrar can read and process them. The default is probably sufficient for most sites. You can, however, raise or lower it as necessary.
MaximumOutstandingRequests	Optional; the default value for this property is 0.  The MaximumOutstandingRequests property is used to limit the incoming traffic in terms of "requests processed". Serves as a hard limit.  The MaximumOutstandingRequests property comprises a number <i>n</i> , where <i>n</i> can be any nonzero value.
MaximumIncomingRequests	Optional; the default value for this property is 0.
ARIsCaseInsensitive	When set to FALSE, requires that you provide exact pathnames with regard to upper and lower case for all objects, subobjects, and properties. The default setting, TRUE, allows you to enter paths such as <b>/rad/serv</b> instead of <b>/Rad/Serv</b> .  <b>Note</b> Prime Access Registrar always authenticates the RADIUS attribute User-Name with regard to upper and lower case, regardless of the setting of this flag.
EnableDiameter	Optional; Either TRUE or FALSE; default is TRUE. Set to True when you want to use the Diameter protocol in Prime Access Registrar.

Table 2-34 BackingStore/ServerParameter Properties (continued)

Fields	Description
<b>KeyStores</b>	
This section is available for each of the following EAP services:	
<ul style="list-style-type: none"> <li>• EAP-SIM</li> <li>• EAP-SIM-3GPP</li> <li>• EAP-AKA</li> <li>• EAP-AKA-3GPP</li> <li>• EAP-AKA-PRIME</li> <li>• EAP-AKA-PRIME-3GPP</li> <li>• EAP-FAST</li> </ul>	
NumberOfKeys	Maximum number of keys stored for generating pseudonym secrets. Value can be from 1 till 1024.
RolloverPeriod	Duration between key updates. Default is 1 week.  In case of rolling encryption, this denotes the duration for which a key is active, after which the key is expired and the next key is considered as an active key for pseudonym generation. For more information on rolling encryption, see <a href="#">Rolling Encryption Support for Pseudonym Generation in EAP-SIM, EAP-AKA, and EAP-AKA' Services</a> , page 5-61.

## Setting Server Parameters

To set up new server parameters:

- 
- Step 1** Choose **Configuration > Advanced > Backing/ServerParam**. The Backing/ServerParam Advanced Details page is displayed.
  - Step 2** Specify the relevant details.
  - Step 3** Click **Set** to save the specified details in the Backing/ServerParamAdvanced Details page.

On successful creation of the server parameters, a success message is displayed else a respective error message is displayed.

---

## RemoteSessionServer

Prime Access Registrar sessions can also be stored on a remote database. This improves the overall scalability of the number of sessions that Prime Access Registrar can simultaneously handle.

The remote session manager internally uses the following two ODBC remote servers:

- Internal-ODBC-Read-Server
- Internal-ODBC-Write-Server

Configurations pertaining to these internal remote servers can be done under the RemoteSessionServer section.

**Note**

Ensure that the length of fields such as Username, Session/Resource Manager name Session-Key, Query-Key and so on are limited to the value specified in the schema, while it is configured. Although the field length of entire session record is 3KB it is limited to 2KB. This is practically sufficient to hold all the session parameters as well as the cached attributes (if any). For more information about the schema, see [Remote Session Management, page 9-48](#).

**Note**

Remote session manager will work only with Oracle database.

[Table 2-35](#) lists and describes the fields in the RemoteSessionServer Advanced Details page.

**Table 2-35 RemoteSessionServer Properties**

Fields	Description
<b>RemoteSessionServer section</b>	
ReactivateTimerInterval	Mandatory time interval (in milliseconds) to activate an inactive server; defaults to 300000 ms.
Timeout	Mandatory time interval (in seconds) to wait for SQL operation to complete; defaults to 15 seconds
DataSourceConnections	Mandatory number of connections to be established; defaults to 8
ODBCDataSource	Name of the ODBCDataSource to use and must refer to one entry in the list of ODBC datasources configured under <b>/Radius/Advanced/ODBCDataSources</b> . Mandatory; no default.
KeepAliveTimerInterval	Mandatory time interval (in milliseconds) to send a keepalive to keep the idle connection active; defaults to zero (0) meaning the option is disabled
MaximumBufferFileSize	Mandatory if BufferAccountingPackets is set to TRUE, determines the maximum buffer file size, defaults to 10 Megabyte)
CacheLimit	Default is 250000; This represents the overall limit on cache of all 'remote' session managers. This value is interpreted as the maximum number of packets that can be present in cache. When the number of sessions hits this limit, sessions will be 'cached out'. This cache out operation will continue, until the cache is at least 20% free.
BufferAccountingPackets	Mandatory, TRUE or FALSE, determines whether to buffer the accounting packets to local file, defaults to TRUE which means that packet buffering is enabled.  <b>Note</b> When set to TRUE, a constant flow of incoming accounting packets can fill the buffer backing store files in <b>/cisco-ar/data/odbc</b> beyond the size configured in MaximumBufferFileSize. Configure BackingStoreDiscThreshold in <b>/Radius/Advanced</b> when using ODBC accounting.

**Table 2-35 RemoteSessionServer Properties (continued)**

Fields	Description
UseCacheIndex	Mandatory; If set to 1, it enables a fast cache based lookup index for the items in the database. This optimizes the number of queries to the database hence will improve performance, but limits the number of sessions that can be scaled.  If set to 0, it disables fast cache based lookup index.
OCITimeOutCount	Required; continuous timeout count to disconnect the selected connection. Default value is 10.
OCIConnectionReactivation-Interval	Required; time interval for attempting to reconnect the disconnected OCI remote server session. Default value is 3000 ms.
OCIActiveConnection-ThresholdCount	Required; threshold count of disconnections after which Prime Access Registrar will mark the remote server as down and try to reactivate it. Default value is 4.

### Setting RemoteSessionServer Details

To set a new RemoteSessionServer details:

- 
- Step 1** Choose **Configuration > Advanced > RemoteSessionServer**. The RemoteSessionServer Advanced Details page appears.
  - Step 2** Specify the relevant details.
  - Step 3** Click **Set** to save the specified details in the RemoteSessionServer Advanced Details page.

On successful creation of the RemoteSessionServer details, a success message is displayed else a respective error message is displayed.

---

### SNMP and Server Monitor

Prime Access Registrar provides SNMP MIB for users of network management systems. The supported MIBs enable the network management station to collect state and statistic information from a Prime Access Registrar server. It enables a standard SNMP management station to check the current state of the server as well as the statistics on each client or each proxy remote server. These messages contain information indicating that either the server was brought up or down or that the proxy remote server is down or has come back online.

[Table 2-36](#) lists and describes the fields in the Advanced Details page.

Table 2-36 *SNMP Properties*

Fields	Description
<b>SNMP Info section</b>	
InputQueueHighThreshold	Percentage that indicates the upper limit of the packet input queue usage. Default is 90.  Prime Access Registrar supports traps to indicate input queue usage. When the input buffer exceeds the given high threshold value, Prime Access Registrar generates a carInputQueueFull trap.
InputQueueLowThreshold	Percentage that indicates the lower limit of the packet input queue usage. Default is 60.  After reaching the high threshold, if the buffer usage drops below a low threshold value, Prime Access Registrar generates a carInputQueueNotVeryFull trap.
DiaInputQueueHighThreshold	Percentage that indicates the maximum number of incoming Diameter packets. Default is 90.  When the input buffer exceeds the given high threshold value, Prime Access Registrar generates a carDiaInputQueueFull trap.
DiaInputQueueLowThreshold	Percentage that indicates the minimum number of incoming Diameter packets. Default is 60.  After reaching the high threshold, if the buffer usage drops below a low threshold value, Prime Access Registrar generates a carDiaInputQueueNotFull trap.
Enabled	Check the box to enable SNMP settings.
TracingEnabled	Check the box to enable all possible tracing in SNMP agent. Tracing is used for debugging purposes.
MasterAgentEnabled	To use SNMP, enable the master agent. Prime Access Registrar responds to SNMP queries through the SNMP master agent.
<b>RFC Compliance Info section</b>	
AllowRejectAttrs	When AllowRejectAttrs is set to FALSE, Reply-Message attributes will not be passed in an Access Reject packet. When AllowRejectAttrs is set to TRUE, attributes will be allowed to pass in an Access Reject packet.
AllowEAPRejectAttrs	When AllowEAPRejectAttrs is set to FALSE, Reply-Message attributes will not be passed in an Access Reject packet if the packet contains EAP-Message attribute. When AllowEAPRejectAttrs is set to TRUE, attributes will be allowed to pass in an Access Reject packet even if the packet contains EAP-Message attribute.
<b>Reply Messages section</b>	
Default	Optional; when you set this property, Cisco Prime Access Registrar sends this value when the property corresponding to the reject reason is not set.

Table 2-36 SNMP Properties (continued)

Fields	Description
UnknownUser	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever Cisco Prime Access Registrar cannot find the user specified by <b>User-Name</b> .
UserNotEnabled	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever the user account is disabled.
UserPasswordInvalid	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever the password in the Access-Request packet did not match the password in the database.
UnableToAcquireResource	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever one of the Resource Managers was unable to allocate the resource for this request.
ServiceUnavailable	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever a service the request needs (such as a RemoteServer) is unavailable.
InternalError	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever an internal error caused the request to be rejected.
MalformedRequest	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever a required attribute (such as <b>User-Name</b> ) is missing from the request.
ConfigurationError	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever the request is rejected due to a configuration error. For example, if a script sets an environment variable to the name of an object such as <b>Authentication-Service</b> , and that object does not exist in the configuration, the reason reported is ConfigurationError.
IncomingScriptFailed	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever one of the <b>IncomingScripts</b> fails to execute.
OutgoingScriptFailed	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever one of the <b>OutgoingScripts</b> fails to execute.
IncomingScriptRejectedRequest	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever one of the <b>IncomingScripts</b> rejects the Access-Request.
TerminationAction	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever Cisco Prime Access Registrar processes the Access-Request as a Termination-Action and is being rejected as a safety precaution.

Table 2-36 *SNMP Properties (continued)*

Fields	Description
OutgoingScriptRejectedRequest	Optional; when you set this property, Cisco Prime Access Registrar sends back this value in the <b>Reply-Message</b> attribute whenever one of the <b>OutgoingScripts</b> rejects the Access-Request.
<b>Server Monitor section</b>	
The following parameters enable monitoring the performance of Prime Access Registrar server.	
TPSHighThreshold	Percentage that indicates the maximum transactions per second (TPS) value for the server. Helps monitoring the TPS capacity of the server. Default is 0.  When the transactions exceed the given high threshold value, Prime Access Registrar generates a carTPSCapacityFull trap.
TPSLowThreshold	Percentage that indicates the minimum TPS value for the server. Helps monitoring the TPS capacity of the server. Default is 0.  After reaching the high threshold, if the TPS value drops below a low threshold value, Prime Access Registrar generates a carTPSCapacityNotFull trap.
SigtranTPSHighThreshold	Percentage that indicates the maximum TPS value for SIGTRAN server. Helps to monitor the TPS capacity of the SIGTRAN server. Default is 0.  When the transactions exceed the given high threshold value, Prime Access Registrar generates a carSigtranTPSCapacityFull trap.
SigtranTPSLowThreshold	Percentage that indicates the minimum TPS value for the SIGTRAN server. Helps to monitor the TPS capacity of the SIGTRAN server. Default is 0.  After reaching the high threshold, if the TPS value drops below a low threshold value, Prime Access Registrar generates a carSigtranTPSCapacityNotFull trap.
SMHighThreshold	Percentage that indicates the maximum number of sessions that can be handled by the server. Default is 0.  When the number of sessions exceeds the given high threshold value, Prime Access Registrar generates a carSessionCapacityFull trap.
SMLowThreshold	Percentage that indicates the minimum number of sessions that can be handled by the server. Default is 0.  After reaching the high threshold, if the number of sessions drops below a low threshold value, Prime Access Registrar generates a carSessionCapacityNotFull trap.
SigtranSMHighThreshold	Percentage that indicates the maximum number of sessions that can be handled by the SIGTRAN server. Default is 0.  When the number of sessions exceeds the given high threshold value, Prime Access Registrar generates a carSigtranSessionCapacityFull trap.

**Table 2-36** *SNMP Properties (continued)*

Fields	Description
SigtranSMLowThreshold	Percentage that indicates the minimum number of sessions that can be handled by the SIGTRAN server. Default is 0.  After reaching the high threshold, if the number of sessions drops below a low threshold value, Prime Access Registrar generates a car-SigtranSessionCapacityNotFull trap.
ServerMonitorLogFreqInsecs	Frequency (in seconds) of monitoring the TPS and sessions.

## Setting SNMP Details

To set up new SNMP details:

**Step 1** Choose **Configuration > Advanced > SNMP**. The SNMP Advanced Details page is displayed.

**Step 2** Specify the relevant details.

**Step 3** Click **Set** to save the specified details in the SNMP Advanced Details page.

On successful creation of the SNMP details, a success message is displayed else a respective error message is displayed.

## DDNS

Prime Access Registrar supports Dynamic DNS Remote server. It is a method, protocol, or network that notifies the server to change the active DNS configuration of its configured hostnames, addresses or other information stored in DNS.

You can click the Add button in the DDNS Details page to enter the TSIGKeys details in the TSIGKeys Details section.

[Table 2-37](#) lists and describes the fields in the TSIGKeys Details section.

**Table 2-37** *TSIGKeys Properties*

Fields	Description
Name	Name of the TSIG Key.
Secret	Set to the same base64-encoded string as defined in the DNS server.
Description	Description of the TSIG Key

You can use the DDNS Details page for the following:

- [Filtering Records](#)
- [Setting DDNS Details](#)
- [Adding the TSIGKeys for DDNS](#)
- [Editing Records](#)
- [Deleting Records](#)

## Setting DDNS Details

To set up new DDNS details:

- 
- Step 1** Choose **Configuration > Advanced > DDNS**. The DDNS Details page is displayed.
- Step 2** Check the **SynthesizeReverseZone** check box, and click **Set DDNS**.
- 

## Adding the TSIGKeys for DDNS

To add TSIGKeys details for DDNS:

- 
- Step 1** Choose **Configuration > Advanced > DDNS**. The DDNS Details page is displayed.
- Step 2** Click **Add**. The TSIGKeys details section is displayed.
- Step 3** Enter the relevant details.
- Step 4** Click **Add** to save the specified details in the TSIGKeys Details section.
- On successful creation of the TSIGKeys details, a success page is displayed else a respective error message is displayed.
- 

## Encrypted IMSI Private Keys

Prime Access Registrar allows you to set up private keys that can help in decrypting an encrypted IMSI from an incoming message for EAP-SIM, EAP-AKA, and EAP-AKA' services.

[Table 2-38](#) lists and describes the fields in the **EncryptedIMSI-PrivateKeys Details** page.

**Table 2-38** Encrypted IMSI-Private Key Details

Fields	Description
AllowedKeyIdentifiers	Allowed key identifier value. This is the key identifier that appears in the incoming EAP response.  Click <b>SetAllowedKeyIdentifiers</b> to set the entered value as the default key identifier.
Name	Name of the private key to map to the key identifier, that can be used to decrypt the incoming encrypted IMSI.
Identifier	The key identifier value.
PrivateKey	The private key value.



### Note

You need to save and reload for the changes to take effect.

You can use the EncryptedIMSI-PrivateKeys Details page for the following:

- [Filtering Records](#)
- [Adding Encrypted IMSI Private Keys](#)

- [Editing Records](#)
- [Deleting Records](#)

## Adding Encrypted IMSI Private Keys

To add private keys for encrypted IMSI:

- 
- Step 1** Choose **Configuration > Advanced > EncryptedIMSIPrivateKeys**. The **EncryptedIMSI-PrivateKeys** page is displayed.
- Step 2** Click **Add** to add new private keys.
- Step 3** Enter the relevant details.
- Step 4** Click **Add** to save the specified details.

The **EncryptedIMSI-PrivateKeys** page is displayed with the newly added keys and a success message is displayed else a respective error message is displayed.




---

**Note** You need to save and reload for the changes to take effect.

---

## ODBC DataSources

Prime Access Registrar uses ODBC as the datasource name to be used by the remote server. Multiple remote servers can use the same ODBCDataSource. Under the ODBCDataSource object definition, a list defines **ODBC.ini** filename/value pairs for a connection. The list includes a Type field and a Driver field, different for each Driver and Data Source, to indicate its Driver and Data Source. Prime Access Registrar supports only the Easysoft Open Source Oracle Driver.

[Table 2-39](#) lists and describes the fields in the Add ODBC DataSources page.

**Table 2-39** *ODBCDataSource Properties*

Fields	Description
Name	Name of the ODBCDataSource
Description	Optional; Description of the ODBC Data Source
Type	Required; type of the ODBC data source, which could be myodbc or oracle_oci.
Driver	Required; <b>liboatodbc.so</b> (default value) <b>Note</b> This attribute is supported only for OBDC.
UserID	Required; database username (no default value)
Password	Optional; user password; shown encrypted
DataBase	Required; Oracle Client configuration database name (no default value)
Server	Set the name of the server
Port	Set the port details.

You can use the ODBC DataSources page for the following:

- [Filtering Records](#)
- [Adding ODBC Data Source](#)
- [Log](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding ODBC Data Source

To add new ODBC data source details:

- 
- Step 1** Choose **Configuration > Advanced > ODBC DataSources**. The ODBC DataSources page is displayed.
- Step 2** Click **Add** to add new ODBC data source details. The ODBC DataSources Details page is displayed.
- Step 3** Enter the relevant details.
- Step 4** Click **Submit** to save the specified details. Otherwise click **Cancel** to return to the ODBC DataSources page without saving the details.

The ODBC DataSources page is displayed with the newly added details and a success message is displayed else a respective error message is displayed.

---

## Log

The log files defined in Prime Access Registrar assist you in identifying the issues related to it. Prime Access Registrar holds sets of log files to store information relevant to server agent processes, monitoring arserver utility, execution of aregcm commands, mcd internal database details, RADIUS server processes and debug details of RADIUS request process.

[Table 2-40](#) lists and describes the fields in the Log Files page.

**Table 2-40 Log Details**

Fields	Description
<b>GUI Log Settings section</b>	
LOG LEVEL	Select either Debug level or Error.
MaxFileSize	Set the maximum size of the log file.
<b>Advance Details section</b>	
LogFileSize	<p>Required; the default is 1 megabyte. This property specifies the maximum size of the RADIUS server log file. The value for the <b>LogFileSize</b> field is a string composed of two parts; a number, and a units indicator (&lt;n&gt; &lt;units&gt;) in which the unit is one of: K, kilobyte, kilobytes, M, megabyte, megabytes, G, gigabyte, or gigabytes.</p> <p>The <b>LogFileSize</b> property does not apply to the <b>config_mcd_1_log</b> or <b>agent_server_1_log</b> files.</p> <p><b>Note</b> This does not apply to the trace log.</p>

Table 2-40 Log Details (continued)

Fields	Description
LogFileCount	Required; the default is 2. This property specifies the number of log files to be kept on the system. A new log file is created when the log file size reaches <b>LogFileCount</b> .  The <b>LogFileCount</b> property does not apply to the <b>config_mcd_1_log</b> or <b>agent_server_1_log</b> files.
TraceFileSize	Required; the default is 1 GB. This property specifies the size of the trace files to be kept on the system. A new trace file is created when the trace file size reaches <b>TraceFileSize</b> . The value for the <b>TraceFileSize</b> field is a string composed of two parts; a number, and a units indicator (<n> <units>) in which the unit is one of: K, kilobyte, kilobytes, M, megabyte, megabytes, G, gigabyte, or gigabytes.
TraceFileCount	Required; this value can be set from 1–100, and the default is 2. This property specifies the number of trace files to maintain. A value of 1 indicates that no file rolling occurs.
LogServerActivity	Required; the default is FALSE, which means Cisco Prime Access Registrar logs all responses except Access-Accepts and Access-Challenges. Accepting the default reduces the load on the server by reducing that amount of information it must log. Note, the client is probably sending accounting requests to an accounting server, so the Access-Accept requests are being indirectly logged. When you set it to TRUE, Cisco Prime Access Registrar logs all responses to the server log file.
TraceLevel	Set the trace level.
LogTPSActivity	When set to TRUE, this property enables to log the TPS usage in a CSV file. The TPS is logged in the following format:  <mm-dd-yyyy>, <hh:mm:ss>, <tps-value>  For example, 04-01-2013, 12:00:01, 102  The default is False.
TPSLogFileCount	Required only if you check the LogTPSActivity check box; the number of TPS Sampling log files to maintain in the repository. The default value is 2.
TPSLogFileNamePrefix	Required only if you check the LogTPSActivity check box; this represents the prefix of the CSV file which will be available in the logs directory of Prime Access Registrar. The following represents the CSV filename format:  <user-prefix>-<mm-dd-yyyy>.csv  tps-04-01-2013.csv
TPSSamplingPeriodInSecs	Required only if you check the LogTPSActivity check box; this represents the TPS sampling period in seconds. The minimum sampling period is set to 5. The default is 30.
EnableSIGTRANStackLogs	When set to TRUE, this property enables to log the SIGTRAN stack logs in stack.log file.

**Table 2-40 Log Details (continued)**

Fields	Description
SIGTRANStackLogFileSize	Required if you check the EnableSIGTRANStackLogs check box. This property specifies the maximum size (in megabyte) of the SIGTRAN stack log file.
SIGTRANLogFileCount	Required if you check the EnableSIGTRANStackLogs check box. This value can be set from 1–100, and the default is 10. This property specifies the number of SIGTRAN log files to maintain in the repository.
LogSessionActivity	When set to TRUE, this property enables Prime Access Registrar to log the session count in the server.
SessionLogFileCount	Required only if you check the LogSessionActivity check box; the number of session log files to maintain in the repository. The default value is 2.
SessionLogFileNamePrefix	Required only if you check the LogSessionActivity check box; this represents the prefix of the session log file which will be available in the logs directory of Prime Access Registrar.
SessionSamplingPeriodInSecs	Required only if you check the LogSessionActivity check box; this represents the session sampling period in seconds. The minimum sampling period is set to 5. The default is 30.

You can use the Log Files page for the following:

- [Filtering Records](#)
- [Viewing Log Details](#)
- [Downloading Log Details](#)
- [Setting Log Details](#)

### Viewing Log Details

To view the log files:

- 
- Step 1** Choose **Configuration > Advanced > Log**. The Log Files page is displayed.
- Step 2** Choose the appropriate radio button and click **View** to view the file.
- 

### Downloading Log Details

To download the log files:

- 
- Step 1** Choose **Configuration > Advanced > Log**. The Log Files page is displayed.
- Step 2** Choose the appropriate radio button and click **Download** to download the file.
-

## Setting Log Details

To set the log details:

- 
- Step 1** Choose **Configuration > Advanced > Log**. The Log Files page is displayed.
- Step 2** Enter the relevant details and click **Set** to save the specified details.
- 

## Ports

The Ports list specifies which ports to listen to for requests. When you specify a port, Prime Access Registrar makes no distinction between the port used to receive Access-Requests and the port used to receive Accounting-Requests. Either request can come in on either port.

Most NASs send Access-Requests to port 1812 and Accounting-Requests to 1813, however, Prime Access Registrar does not check.

When you do not specify any ports, Prime Access Registrar reads the /etc/services file for the ports to use for access and accounting requests. If none are defined, Prime Access Registrar uses the standard ports (1812 and 1813).

[Table 2-41](#) lists and describes the fields in the Ports page.

**Table 2-41 Port Properties**

Fields	Description
Port	Required; allows you to use ports other than the default, 1812 and 1813. You can use this option to configure Prime Access Registrar to use other ports,. If you add additional ports, however, Prime Access Registrar will use the added ports and no longer use the default ports 1812 and 1813. These default ports can still be used by adding them to the list of ports to use.
Type	Set the port type.
Description	Optional; description of the port.

You can use the Ports page for the following:

- [Filtering Records](#)
- [Adding Port Details](#)
- [Interfaces](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Port Details

To add new port details:

- 
- Step 1** Choose **Configuration > Advanced > Port**. The Ports page is displayed.

- Step 2** Enter the relevant details and click **Add**. The new port details will be listed in the Ports page.
- 

## Interfaces

The Interfaces list specifies the interfaces on which the RADIUS server receives and sends requests. You specify an interface by its IP address.

- When you set an IP address, Prime Access Registrar uses that interface to send and receive Access-Requests.
- When no interfaces are listed, the server performs an interface discover and uses all interfaces of the server, physical and logical (virtual).



**Note**

The IP address format is enhanced to support both IPv4 and IPv6.

---

You can use the interfaces page for the following:

- [Filtering Records](#)
- [Adding IP Addressing Interface](#)
- [Deleting Records](#)

### Adding IP Addressing Interface

To add a new IP address interface to define an interface:

---

- Step 1** Choose **Configuration > Advanced > Interfaces**. The Interfaces page is displayed.

- Step 2** Enter the **IP Address** and click **Add**.

The Interfaces page is displayed with the newly added details and a success message is displayed else a respective error message is displayed.

---

## Attribute Groups

The Attributes can be grouped using Prime Access Registrar Profile object. The attributes for a particular user group can be grouped under a profile and the attributes contained in the profiles will be returned in their access-accepts.

[Table 2-42](#) lists and describes the fields in the Attribute Groups Details page.

**Table 2-42** *AttributeGroups Properties*

Fields	Description
Name	Name of the attribute group.
Description	Optional; description of the attribute group.

**Table 2-42** *AttributeGroups Properties (continued)*

Fields	Description
Attribute type	Select either <b>RADIUS</b> or <b>VENDOR</b> . If Vendor is selected, specify the vendor type from the drop-down list.
Attribute Name	Optional; based on the Attribute Type selected, the attribute name is automated. Set the relevant name for the attribute type selected. Click the <b>Add</b> button to save the details and list it in Attribute list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.

You can use the Attribute Groups page for the following:

- [Filtering Records](#)
- [Adding Attribute Group Details](#)
- [Rules](#)
- [Editing Records](#)
- [Deleting Records](#)

### Adding Attribute Group Details

To add new attribute groups details:

- 
- Step 1** Choose **Configuration > Advanced > Attributes Groups**. The Attribute Groups page is displayed.
  - Step 2** Click **Add** to add new attribute group details. The Attribute Group Details page is displayed.
  - Step 3** Enter the relevant details.
  - Step 4** Click **Submit** to save the specified details in the Attribute Groups Details page. Otherwise click **Cancel** to return to the Attribute Groups page without saving the details.

The Attribute Groups page is displayed with the newly added details or a respective error message is displayed.

---

## Rules

A Rule is a function that selects services based on all input information used by the function.

[Table 2-43](#) lists and describes the fields in the Add Rules List page.

**Table 2-43** *Rule Properties*

Fields	Description
<b>General Properties tab</b>	
Name	Required; must be unique in the Rule list.

**Table 2-43 Rule Properties (continued)**

Fields	Description
Description	Optional; description of the rule.
Type	Required; specifies the type of the rule which can be Radius or Diameter.
Script Name	Name of the script.
<b>Attribute Details tab</b>	
These fields are displayed based on the type of the rule selected in the Type field.	
RADIUS	Optional; set Radius, if the attribute and value need to be defined for RADIUS.
VENDOR	Optional; set Vendor, if the attribute and value need to be defined for Vendor.
AttributeName	Optional; based on the Attribute Type selected, the attribute name is automated. Set the relevant name for the attribute type selected.
AttributeValue	Optional; set the value for the selected attribute. Click the <b>Add</b> button to save the details and list it in Name and Value list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.

You can use the Rules List page for the following:

- [Filtering Records](#)
- [Setting Rules](#)
- [SessionManagers](#)
- [Editing Records](#)
- [Deleting Records](#)

## Setting Rules

To set new rules:

- 
- Step 1** Choose **Configuration > Rules**. The List of Rules page is displayed.
  - Step 2** Click **Add**. The Rules Details page is displayed.
  - Step 3** Enter the relevant details.
  - Step 4** Click **Submit** to save the specified details in the Rules Details page. Otherwise click **Cancel** to return to the List of Rules page without saving the details.

The List of Rules page is displayed with the newly added details or a respective error message is displayed.

---

## SessionManagers

You can use Session Managers to track user sessions. The Session Managers monitor the flow of requests from each NAS and detect the session state. When requests come through to the Session Manager, it creates sessions, allocates resources from appropriate Resource Managers, and frees and deletes sessions when users log out.

The Session Manager enables you to allocate dynamic resources to users for the lifetime of their session. You can define one or more Session Managers and have each one manage the sessions for a particular group or company.


**Note**

Session record size is limited by the operating system (OS) paging size (4 KB in Linux). If a request triggers creation of a session that exceeds the OS paging size, the request will be dropped and the session will not be created.


**Note**

In this release of Prime Access Registrar, the memory capacity is enhanced to store more than 4 million active session's by storing the active session records in database server instead of storing it in the main memory. The capacity is dependent on the number of attributes that are being captured for each session.


**Note**

If the disk partition where Prime Access Registrar stores session backing store data (usually the disk partition where Prime Access Registrar is installed, such as `/opt/CSCOar`) is full, the subsequent packets that try to create sessions will be dropped and no sessions will be created due to lack of disk space.

Session Managers use Resource Managers, which in turn, manage a pool of resources of a particular type.

[Table 2-44](#) lists and describes the fields in the Session Manager Details page.

**Table 2-44**      **Session Manager Properties**

Fields	Description
Name	Required; must be unique in the Session Managers list.
Description	Optional description of the Session Manager.
Type	Required; set to local or remote. Local is the traditional session manager that maintains sessions in memory and has good performance. The remote session manager operates on a remote ODBC database, and its performance is highly dependent on the performance of the ODBC database.
EnableDiameter	Optional; check the box if you want to use the session manager for Diameter services.

**Table 2-44 Session Manager Properties (continued)**

Fields	Description
SessionKey	<p>SessionKey property is used to set the sessionkey value for the Session Manager.</p> <p>The SessionManager checks whether the environmental variable <b>Session-Key</b> is set or not. If the environmental variable is set, the server uses it as the sessionkey. If environmental variable <b>Session-Key</b> is not set then SessionManager gets the value configured in the SessionKey property under SessionManager.</p> <p>SessionKey can be a combination of attributes separated by a colon. The values for those attributes are obtained from the RequestDictionary. If any one of the attribute that is configured for the sessionkey is not present in the RequestDictionary, Prime Access Registrar will drop the request.</p> <p>However, if <b>Session-Key</b> is not set, SessionManager uses NAS-Identifier and NAS-Port to create the sessionkey. An example configuration,</p> <pre>--&gt; set SessionKey "User-Name:NAS-Port"</pre> <p>The following shows the sample configuration of sessionkey for Session Manager:</p> <pre>[ //localhost/Radius/SessionManagers/session-mgr-1 ] Name = session-mgr-1 Description = Type = local EnableDiameter = FALSE IncomingScript = OutgoingScript = AllowAccountingStartToCreateSession = TRUE SessionTimeOut = PhantomSessionTimeOut = SessionKey = ResourceManagers/</pre>
AllowAccountingStartToCreateSession	<p>Set to TRUE by default; start the session when the Prime Access Registrar server receives an Access Accept or an Accounting-Start.</p> <p>When set to FALSE, start the session when the Prime Access Registrar server receives an Access Accept.</p>
IncomingScript	Optional; name of script to run when the service starts. This script is run as soon as the session is acquired in Prime Access Registrar.
OutgoingScript	Optional; script to be run just before the session is written to backing store.

Table 2-44 Session Manager Properties (continued)

Fields	Description
SessionTimeout	<p>The SessionTimeout property is optional; no value for this property means the session timeout feature is disabled.</p> <p>Used in conjunction with <b>/Radius/Advanced/SessionPurgeInterval</b> for the session timeout feature. Enables the session timeout feature for a Session Manager. If the SessionTimeout property is set to a value under a session manager, all sessions that belong to that session manager will be checked for timeouts at each SessionPurgeInterval. If any sessions have timed out, they will be released, and all resources associated with those sessions are also released.</p> <p>The SessionTimeout property determines the timeout for a session. If the time difference between the current time and the last update time is greater than this property's value, the session is considered to be stale. The last update time of the session is the time at which the session was created or updated.</p> <p>The SessionTimeout value is comprised of a number and a units indicator, as in <i>n units</i>, where a unit is one of minutes, hours, days, or weeks. The default unit is 'days'.</p>
PhantomSessionTimeout	<p>Optional; no value for this property means the phantom session timeout feature is disabled.</p> <p>The PhantomSessionTimeout property is used in conjunction with <b>/Radius/Advanced/SessionPurgeInterval</b> to enable the phantom session timeout feature for Session Manager.</p> <p>If the PhantomSessionTimeout property is set to a value under a session manager, all sessions that belong to that session manager will be checked for receipt of an Accounting-Start packet. Sessions that do not receive an Accounting-Start packet from creation until its timeout will be released.</p> <p>The PhantomSessionTimeout value comprises a number and a units indicator, as in <i>n units</i>, where a unit is one of minutes, hours, days, or weeks. The default unit is 'days'.</p>
SessionCreationCmdList	Available only if you check the EnableDiameter check box; session created for the configured application, command code, and AVP.
SessionDeletionCmdList	Available only if you check the EnableDiameter check box; session deleted for the configured application, command code, and AVP.

**Table 2-44 Session Manager Properties (continued)**

Fields	Description
SessionRestorationTime-out	<p>Determines the restoration timeout for a session. No value indicates that the session restoration feature is disabled for this session manager. Used in conjunction with <b>/Radius/Advanced/DiameterSessionRestorationPurgeTime</b>.</p> <p>This value comprises a number and a units indicator, as in ‘n’ units, where a unit could be minutes, hours, days, or weeks. The default unit is ‘days’. The minimum recommended value is <b>24hr</b> or <b>1Day</b>.</p> <p>If this value is set for a session manager, all sessions that belong to that session manager will be checked for timeouts at DiameterSessionRestorationPurgeTime. If any session is timed out, a Re-Authorization-Request will be triggered for the timed-out session. And, if Re-Authorization-Answer comes with the Result-Code Diameter-Unknown-Session-Id, then the particular session will be released and all resources associated with the session will also be released.</p> <p>If the time difference between the current time and the last update time for the session is greater than this value, the session is considered to be stale and must be restored.</p> <p><b>Note</b> Session restoration works only if the session manager is Diameter enabled and it has a 3GPP resource manager.</p>
Resource Managers List	Ordered list of Resource Managers. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details.

You can use the Session Managers page for the following:

- [Filtering Records](#)
- [Adding Session Manager Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Session Manager Details

To add new session manager details:

- 
- Step 1** Choose **Configuration > Session Managers**. The Session Managers page is displayed.
  - Step 2** Click **Add**. The Session Manager Details page is displayed.
  - Step 3** Enter the required details.
  - Step 4** Click **Add** to save the specified details in the Session Manager Details page. Otherwise click **Cancel** to return to the Session Managers page without saving the details.

The Session Managers page is displayed with the newly added details or a respective error message is displayed.

---

## ResourceManager

Resource Managers allow you to allocate dynamic resources to user sessions. The following lists the different types of Resource Managers.

- **IP-Dynamic**—manages a pool of IP addresses that allows you to dynamically allocate IP addresses from a pool of addresses
- **IP-Per-NAS-Port**—allows you to associate ports to specific IP addresses, and thus ensure each NAS port always gets the same IP address
- **IPX-Dynamic**—manages a pool of IPX network addresses
- **Subnet-Dynamic**—manages a pool of subnet addresses
- **Group-Session-Limit**—manages concurrent sessions for a group of users; that is, it keeps track of how many sessions are active and denies new sessions after the configured limit has been reached
- **User-Session-Limit**—manages per-user concurrent sessions; that is, it keeps track of how many sessions each user has and denies the user a new session after the configured limit has been reached
- **Home-Agent**—manages a pool of on-demand IP addresses
- **USR-VPN**—manages Virtual Private Networks (VPNs) that use USR NAS Clients.
- **Home-Agent-IPv6**—manages a pool of on-demand IPv6 addresses
- **Remote-IP-Dynamic**—manages a pool of IP addresses that allows you to dynamically allocate IP addresses from a pool of addresses. It internally works with a remote ODBC database.
- **Remote-User-Session-Limit**—manages per-user concurrent sessions; that is, it keeps track of how many sessions each user has and denies the user a new session after the configured limit has been reached. It internally works with a remote ODBC database.
- **Remote-Group-Session-Limit**—manages concurrent sessions for a group of users; that is, it keeps track of how many sessions are active and denies new sessions after the configured limit has been reached. It internally works with a remote ODBC database.
- **Session Cache**—allows you to define the RADIUS attributes to store in cache.
- **Dynamic-DNS**—manages the DNS server.
- **Remote-Session-Cache**—allows you to define the RADIUS attributes to store in cache. It should be used with session manager of type 'remote'.
- **3GPP**—allows you to define the attribute for 3GPP authorization.

Each Resource Manager is responsible for examining the request and deciding whether to allocate a resource for the user, do nothing, or cause Cisco Prime Access Registrar to reject the request.

[Table 2-45](#) lists and describes the fields in the Resource Manager Details page.

**Table 2-45 Resource Manager Properties**

Fields	Description
Resource Manager Name	Required; must be unique in the Resource Managers list.

**Table 2-45** Resource Manager Properties (continued)

Fields	Description
Description (optional)	Optional; description of the Resource Manager.
Type	Required; must be either <b>Dynamic-DNS</b> , <b>IP-Dynamic</b> , <b>IP-Per-NAS-Port</b> , <b>IPX-Dynamic</b> , <b>Session Cache</b> , <b>Subnet-Dynamic</b> , <b>Group-Session-Limit</b> , <b>Home-Agent</b> , <b>User-Session-Limit</b> , <b>USR-VPN</b> , <b>Home-Agent-IPv6</b> , <b>Remote-IP-Dynamic</b> , <b>Remote-User-Session-Limit</b> , <b>Remote-Group-Session-Limit</b> , <b>Remote-Session-Cache</b> , or <b>3GPP</b> . Based on the option selected, the fields displayed in the Resource Manager Details page varies.

The fields displayed in the Resource Manager Details page changes based on the option selected in the Type field. The following tables describe the fields in the Resource Manager Details page.

### DYNAMIC-DNS

[Table 2-46](#) lists and describes the fields in the Resource Manager Details page.

**Table 2-46** DYNAMIC-DNS Properties

Fields	Description
<b>General tab</b>	
Max DNS TTLS	Set the maximum TTL of the DNS record.
DNS Host bytes	Set the number of bytes to be used to construct the reverse zone entry.
Forward Zone Name	Set the name of the forward zone. For a given Resource Manager you must decide which forward zone you will be updating for sessions the resource manager will manage.
Reverse Zone Name	Set the name of the reverse zone.
Forward Zone Server	Set the Server IP of the forward zone
Reverse Zone Server	Set the Server IP of the reverse zone
Forward Zone TSIG KeyS	Server-wide security key to process all forward zone dynamic DNS updates. This is used if a ForwardZoneTSIGKey was not specified on the Resource Manager.
Reverse Zone TSIG Keys	Server-wide security key to process all reverse zone dynamic DNS updates. This is used if a ReverseZoneTSIGKey was not specified on the Resource Manager

### GROUP-SESSION-LIMIT

[Table 2-47](#) lists and describes the fields in the Resource Manager Details page.

**Table 2-47** GROUP-SESSION-LIMIT Properties

Fields	Description
Group Session Limit	Set the GroupSessionLimit property to the maximum number of concurrent sessions for all users.

**REMOTE-GROUP-SESSION-LIMIT**

Table 2-48 lists and describes the fields in the Resource Manager Details page.

**Table 2-48** *REMOTE-GROUP-SESSION-LIMIT Properties*

Fields	Description
Group Session Limit	Set the GroupSessionLimit property to the maximum number of concurrent sessions for all users.

**HOME-AGENT**

Table 2-49 lists and describes the fields in the Resource Manager Details page.

**Table 2-49** *HOME-AGENT Properties*

Fields	Description
<b>HomeAgentIPAddresses tab</b>	
Start	Required; must be an IP address.
End	Required; must be an IP address.

Click the **Add** button to save the details and list it in Start and End IP list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See [Relocating Records](#) for more details. To delete the available attributes, select the relevant attribute and click the **Delete** button below.

**HOME-AGENT-IPv6**

Table 2-50 lists and describes the fields in the Resource Manager Details page.

**Table 2-50** *HOME-AGENT-IPv6 Properties*

Fields	Description
<b>HomeAgentIPv6Addresses tab</b>	
Start	Required; must be an IPv6 address.
End	Required; must be an IPv6 address.

Click the **Add** button to save the details and list it in Start and End IPv6 list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See [Relocating Records](#) for more details. To delete the available attributes, select the relevant attribute and click the **Delete** button below.

**IP-DYNAMIC**

Table 2-51 lists and describes the fields in the Resource Manager Details page.

**Table 2-51** *IP-DYNAMIC Properties*

Fields	Description
<b>General tab</b>	
Reuse IP for same SessionKey and User	When set to TRUE, this property supports overlapping IP addresses between session managers for VPN users. Default value is FALSE.
Net Mask	Required; must be set to a valid net mask.

**Table 2-51** *IP-DYNAMIC Properties (continued)*

Fields	Description
Allow Overlapped IP Addresses	When set to TRUE, this property supports overlapping IP addresses between session managers for VPN users. Default value is FALSE.
<b>IP Addresses tab</b>	
Start	Required; must be an IP address.
End	Required; must be an IP address.

Click the **Add** button to save the details and list it in Start and End IP list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See [Relocating Records](#) for more details. To delete the available attributes, select the relevant attribute and click the **Delete** button below.

### REMOTE-IP-DYNAMIC

[Table 2-52](#) lists and describes the fields in the Resource Manager Details page.

**Table 2-52** *REMOTE-IP-DYNAMIC Properties*

Fields	Description
<b>General tab</b>	
Reuse IP for same SessionKey and User	When set to TRUE, this property supports overlapping IP addresses between session managers for VPN users. Default value is FALSE.
Net Mask	Required; must be set to a valid net mask.
Allow Overlapped IP Addresses	When set to TRUE, this property supports overlapping IP addresses between session managers for VPN users. Default value is FALSE.
<b>IP Addresses tab</b>	
Start	Required; must be an IP address.
End	Required; must be an IP address.

Click the **Add** button to save the details and list it in Start and End IP list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See [Relocating Records](#) for more details. To delete the available attributes, select the relevant attribute and click the **Delete** button below.

### IP-PER-NAS-PORT

[Table 2-53](#) lists and describes the fields in the Resource Manager Details page.

**Table 2-53** *IP-PER-NAS-PORT Properties*

Fields	Description
<b>General tab</b>	
Net Mask	Required; if used, must be set to a valid net mask.
Allow Overlapped IP Addresses	When set to TRUE, this property supports overlapping IP addresses between session managers for VPN users. Default value is FALSE.
NAS	Required; must be the name of a known Client. This value must be the same as the NAS-Identifier attribute in the Access-Request packet.

**Table 2-53** *IP-PER-NAS-PORT Properties (continued)*

Fields	Description
<b>IP Config tab</b>	
Start	Required; must be an IP address.
End	Required; must be an IP address.
<b>Port Config tab</b>	
Start	Required; set the NAS port
End	Required; set the NAS port

Click the **Add** button to save the details and list it in Start and End IP list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See [Relocating Records](#) for more details. To delete the available attributes, select the relevant attribute and click the **Delete** button below.

**IPX-DYNAMIC**

[Table 2-54](#) lists and describes the fields in the Resource Manager Details page.

**Table 2-54** *IPX-DYNAMIC Properties*

Fields	Description
<b>Networks tab</b>	
Start	Required; must be an IP address.
End	Required; must be an IP address.

Click the **Add** button to save the details and list it in Start and End IP list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See [Relocating Records](#) for more details. To delete the available attributes, select the relevant attribute and click the **Delete** button below.

**SESSION-CACHE**

[Table 2-55](#) lists and describes the fields in the Resource Manager Details page.

**Table 2-55** *SESSION-CACHE Properties*

Fields	Description
<b>General tab</b>	
Overwrite Attributes	Specifies whether to overwrite the existing attributes if there are any in the session record.
Query Key	<p>Required; set the QueryKey to the a RADIUS attribute you want to key on, such as Framed-IP-Address.</p> <p>A change made in Prime Access Registrar requires that this attribute not be an XML attribute, even if this session-cache resource manager is being used for an XML query.</p> <p><b>Note</b> Any existing session-cache resource managers using an XML attribute for the Query Key must be changed to a RADIUS attribute that this XML attribute is mapped to under Query-Mappings.</p>

**Table 2-55** *SESSION-CACHE Properties (continued)*

Fields	Description
Pending Removal Delay	Required; length of time information remains in the cache after the session ends (defaults to 10 seconds)
<b>Query Mapping tab</b>	
XML Attribute	Set the QueryKey property to the XML attribute you want to key on such as XML-Address-format-IPv4 and list all attributes to be cached in the AttributesToBeCached subdirectory.
Radius Attribute	Required; list of attribute pairs, mapping the XML attributes on the left-hand side to the RADIUS attribute on the right-hand side.
<b>AttributeToBeCached tab</b>	
RADIUS	Optional; set Radius, if the attribute needs to be defined for RADIUS.
VENDOR	Optional; set Vendor, if the attribute needs to be defined for Vendor. If Vendor is selected, specify the vendor type from the drop-down list.
Attribute Name	Required; use this subdirectory to provide a list of RADIUS attributes you want to store in cache

Click the **Add** button to save the details and list it in Start and End IP list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See [Relocating Records](#) for more details. To delete the available attributes, select the relevant attribute and click the **Delete** button below.

### SUBNET-DYNAMIC

[Table 2-56](#) lists and describes the fields in the Resource Manager Details page.

**Table 2-56** *SUBNET-DYNAMIC Properties*

Fields	Description
<b>Subnet Dynamic tab</b>	
Net Mask	Required; must be set to the size of the managed subnets
Start	Required; must be an IP addresses
End	Required; must be an IP addresses

Click the **Add** button to save the details and list it in Start and End IP list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See [Relocating Records](#) for more details. To delete the available attributes, select the relevant attribute and click the **Delete** button below.

### USER-SESSION-LIMIT

[Table 2-57](#) lists and describes the fields in the Resource Manager Details page.

**Table 2-57** *USER-SESSION-LIMIT Properties*

Fields	Description
User Session Limit	Set the user session limit property to the maximum number of concurrent sessions for a particular user

**REMOTE-USER-SESSION-LIMIT**

Table 2-58 lists and describes the fields in the Resource Manager Details page.

**Table 2-58 REMOTE-USER-SESSION-LIMIT Properties**

Fields	Description
User Session Limit	Set the user session limit property to the maximum number of concurrent sessions for a particular user

**USR-VPN**

Table 2-59 lists and describes the fields in the Resource Manager Details page.

**Table 2-59 USR-VPN Properties**

Fields	Description
<b>General tab</b>	
Identifier	Required; must be set to the VPN ID the USR NAS will use to identify a VPN.
Neighbor	Optional; if set, should be the IP address of the next hop router for the VPN.
Framed Routing	Optional; if set, should be <b>RIP V2 Off</b> or <b>RIP V2 On</b> if the USR NAS is to run RIP Version 2 for the user.
<b>Gateway tab</b>	
Name of Gateway	Required; name of the gateway.
Description (optional)	Optional; description of the gateway.
IP Address	Required; IP address of the gateway
Shared Secret	Required; must match the shared secret of the gateway.
Tunnel Refresh	Optional; if specified it is the number of seconds the tunnel stays active before a secure “keepalive” is exchanged between the tunnel peers in order to maintain the tunnel open.
Location ID	Optional; if specified it is a string indicating the physical location of the gateway. Click the <b>Save</b> button, to save the details.

To edit the gateway details, check the appropriate check box and click the **Edit** button. Enter new information in the editable fields and click the **Save** button. You can also delete the record using **Delete** button.

**REMOTE-SESSION-CACHE**

Table 2-60 lists and describes the fields in the Resource Manager Details page.

**Table 2-60 REMOTE-SESSION-CACHE Properties**

Fields	Description
<b>General tab</b>	
Overwrite Attributes	Specifies whether to overwrite the existing attributes if there are any in the session record.

**Table 2-60** *REMOTE-SESSION-CACHE Properties (continued)*

Fields	Description
Query Key	Required; set the QueryKey to the a RADIUS attribute you want to key on, such as Framed-IP-Address.  A change made in Prime Access Registrar requires that this attribute not be an XML attribute, even if this session-cache resource manager is being used for an XML query.  <b>Note</b> Any existing session-cache resource managers using an XML attribute for the Query Key must be changed to a RADIUS attribute that this XML attribute is mapped to under Query-Mappings.
Pending Removal Delay	Required; length of time information remains in the cache after the session ends (defaults to 10 seconds)
<b>Remote Query Mapping tab</b>	
XML Attribute	Set the QueryKey property to the XML attribute you want to key on such as XML-Address-format-IPv4 and list all attributes to be cached in the AttributesToBeCached subdirectory.
Radius Attribute	Required; list of attribute pairs, mapping the XML attributes on the left-hand side to the RADIUS attribute on the right-hand side.
<b>RemoteAttributeToBeCached tab</b>	
RADIUS	Optional; set Radius, if the attribute needs to be defined for RADIUS.
VENDOR	Optional; set Vendor, if the attribute needs to be defined for Vendor. If Vendor is selected, specify the vendor type from the drop-down list.
Attribute Name	Required; use this subdirectory to provide a list of RADIUS attributes you want to store in cache

**3GPP**

[Table 2-61](#) lists and describes the 3GPP properties in the Resource Manager Details page.

**Table 2-61** *3GPP Properties*

Fields	Description
EnableRegistrationFlow	Check the box to enable initiation of a Server-Assignment-Request (SAR) registration message when a session is created and a SAR deregistration message when a session is deleted.
EnableSessionTermination	Check the box to enable initiation of a Server-Termination-Request (STR) message when a session is deleted.
ReuseExistingSession	If selected, SAR registration will not be initiated for an existing session.
HSSProxyService	Required; a service of type Diameter used to group a list of HSS/Diameter servers towards which the SAR and STR messages need to be initiated in the 3GPP authorization flow.

You can use the Resource Manager List page for the following:

- [Filtering Records](#)
- [Adding Resource Manager Details](#)
- [Network Resources](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Resource Manager Details

To add new resource manager details:

- 
- Step 1** Choose **Configuration > Resource Manager**. The Resource Manager List page is displayed.
  - Step 2** Click **Add**. The Resource Manager Details page is displayed.
  - Step 3** Enter the required details.
  - Step 4** Click **Submit** to save the specified details in the Resource Manager Details page. Otherwise click **Cancel** to return to the Resource Manager List page without saving the details.

The Resource Manager List page is displayed with the newly added details or a respective error message is displayed.

---



### Note

Resource Manager supports the following remote type session managers: remote-ip-dynamic, remote-session-cache, home-agent, remote-user-session-limit, home-agent-ipv6 and remote-group-session-limit.

---

## Network Resources

Network Resources constitutes the maintenance and management of the details of the clients and remote servers. The clients IP address and shared secret details are maintained under clients, The management of server directory with use of remote server protocols details are maintained in remote server.

This section describes the following:

- [Clients](#)
- [Remote Servers](#)

## Clients

All NASs and proxy clients that communicate directly with Prime Access Registrar must have an entry in the Clients list. This is required because NAS and proxy clients share a secret with the RADIUS server which is used to encrypt passwords and to sign responses.

[Table 2-62](#) lists and describes the fields in the Client Details page.

**Table 2-62 Client Properties**

Fields	Description
Name	Required and should match the Client identifier specified in the standard RADIUS attribute, <b>NAS-Identifier</b> . The name must be unique within the Clients list.
IncomingScript	Optional; you can use this property to specify a Script you can use to determine the services to use for authentication, authorization, and/or accounting.
OutgoingScript	Optional; you can use this property to specify a Script you can use to make any Client-specific modifications when responding to a particular Client.
Protocol	Required; set it to <b>Radius</b> , <b>Diameter</b> , <b>Radius-TLS</b> , or <b>Tacacs-and-Radius</b> .
Description	Optional description of the client.
Vendor	Optional; displayed when the protocol is set to Diameter. When set, must be the name of a known Vendor.
Server Identity	Optional; displayed when the protocol is set to Diameter. While exchanging the CER information in the client, Prime Access Registrar sends the configured server identity value as the origin-host value. When set, it takes precedence over the /Radius/Advance/Diameter/TransportManagement configuration.
HostName	Required; hostname or IP address of the Diameter client.
Port	Required; port on which client connects with the Prime Access Registrar server.
SCTP-Enabled	Required; displays when the protocol is set to Diameter and indicates whether the connection will be an SCTP. If set to TRUE, SCTP will be used. If set to FALSE, TCP will be used.
Advertised-Realm	Optional; displays when the protocol is set to Diameter. While exchanging the CER information in the client, Prime Access Registrar sends the configured server realm value as the origin-realms value. It takes precedence over the /Radius/Advance/Diameter/TransportManagement configuration.
WatchDogTimeout	Time interval between watch dog messages.
MaxIncomingRequestRate	Maximum number of incoming requests allowed per second.
KeepAliveTime	Time interval, in milliseconds, to keep an idle session active.
InitialTimeout	Timeout value, in milliseconds, the Prime Access Registrar server waits for a response before dropping the packet.
TLS-Enabled	Check this box to enable TLS security mechanism for the Diameter client.
Advertised-HostName	Optional; specifies the local hostname address that will be advertised by the Prime Access Registrar server to other peers during CER/CEA exchange.
AuthSession-StateInASR	When EnableAuthSessionState is set to: <ul style="list-style-type: none"> <li>• No-State-Maintained—When RTR is received from HSS , Auth-Session-State AVP should be set with No-State-Maintained on sending ASR to the client; and the session is deleted.</li> <li>• State-Maintained—When RTR is received from HSS , Auth-Session-State AVP should be set with State-Maintained on sending ASR to client. The session is deleted only on reception of STR from client.</li> </ul>

**Table 2-62 Client Properties (continued)**

Fields	Description
UserLogEnabled	This field is available for protocol of type <b>Diameter</b> . Check this box to display the user information in the log file for example username, AAAID, client identifier, result-code, and diameter-message-type.  If this option is enabled, Prime Access Registrar stores all subscriber messages including Diameter request and response in a separate log file called subscriber_log in the \$INSTALLPATH/logs folder.
MaximumTLS-Connections	This field is available for protocol of type <b>radius-tls</b> .  Maximum number of TLS connections that the client can establish with Prime Access Registrar. Default value is one. Maximum number of TLS connections allowed per client is 50.
<b>SCTPParameters Section</b>	
This section is available if the Sctp-Enabled option is checked.	
SourcePort	Client source port. Default value is 3868.
DestinationPort	Client destination port. Default value is 3868.
PathMaxRetrans	Maximum number of consecutive retransmissions over a destination transport address of a peer endpoint before it is marked as inactive. Default value is 5.
RTOInitial	Initial value of RTO (retransmission timeout) that is used in RTO calculations. Measured in milliseconds and default value is 3 seconds.
RTOCookieLife	Maximum lifespan of the cookie sent in an INIT ACK chunk. Measured in milliseconds and default value is 60 secs.
RTOMin	Minimum value of RTO. Measured in milliseconds and default value is 1 second.
HBInterval	Interval when a HEARTBEAT chunk is sent to a destination transport address to monitor the reachability of an idle destination transport address. Measured in milliseconds and default is 30 seconds.
RTOMax	Maximum value of RTO. Measured in milliseconds and default value is 60 seconds.
SACKTimeout	Delayed SACK timeout. Default value is 200 msec.
MaxInitRetransmits	Maximum number of times an INIT chunk or a COOKIE ECHO chunk is retransmitted before an endpoint aborts the initialization process and closes the association. Default value is 8.
InitNumOSTreams	Initial number of streams per socket.
Association-MaxRetrans	Maximum number of consecutive retransmissions to a peer before an endpoint considers that the peer is unreachable and closes the association. Default value is 10.
InitMaxInStreams	Maximum number of inbound streams per socket.

**Table 2-62 Client Properties (continued)**

Fields	Description
SCTPAdvertisedHostName	Displays set of IP addresses for local and remote hosts.
<b>TLSOptions / RTLS Options</b>	
This section is available if the protocol is set to one of the following: <ul style="list-style-type: none"> <li>• Diameter and TLS-Enabled option is checked</li> <li>• radius-tls</li> </ul>	
PrivateKeyPassword	The password used to protect the server's private key.
ServerKeyFile	<p>The full pathname of the file containing the server's RSA private key. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The valid encoding prefix is "PEM". If an encoding prefix is not present, the file is assumed to be in PEM format.</p> <p>The following example assumes that the subdirectory <b>pki</b> under <b>/cisco-ar</b> contains the server's certificate file. The file <b>server-key.pem</b> is assumed to be in PEM format. The file extension <b>.pem</b> is not significant.</p> <p style="text-align: center;"><b>set ServerKeyFile PEM:/cisco-ar/pki/server-key.pem</b></p>
ServerCertificateFile	The full pathname of the file containing the server's certificate or certificate chain used during the TLS exchange. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The valid encoding prefix is PEM. If an encoding prefix is not present, the file is assumed to be in PEM format.
CACertificateFile	The full pathname of the file containing trusted CA certificates used for client verification. The file can contain more than one certificate, but all certificates must be in PEM format. DER encoding is not allowed.
CACertificatePath	<p>The name of a directory containing trusted CA certificates (in PEM format) used for client verification. This parameter is optional, and if it is used there are some special preparations required for the directory it references.</p> <p>Each certificate file in this directory must contain exactly one certificate in PEM format. The server looks up the certificate files using the MD5 hash value of the certificate's subject name as a key. The directory must therefore also contain a set of symbolic links each of which points to an actual certificate file. The name of each symbolic link is the hash of the subject name of the certificate.</p> <p>For example, if a certificate file named <b>ca-cert.pem</b> is located in the CACertificatePath directory, and the MD5 hash of the subject name contained in <b>ca-cert.path.pem</b> is 1b96dd93, then a symbolic link named 1b96dd93 must point to <b>ca-cert.pem</b>.</p> <p>If there are subject name collisions such as multiple certificates with the same subject name, each link name must be indexed with a numeric extension as in 1b96dd93.0 and 1b96dd93.1.</p>

Table 2-62 Client Properties (continued)

Fields	Description
PeerVerificationMode	Select one of the following options: <ul style="list-style-type: none"> <li>None—If the peer verification certificate must not be requested.</li> <li>Optional—If peer verification certificate can be requested; but, verification is not required.</li> <li>RequireCertificate—If peer certificate must be requested and verified.</li> </ul>
Verification-Depth	Specifies the maximum length of the certificate chain used for client verification.
EnableAuto-Chaining	When set to TRUE, Prime Access Registrar sends its server certificate chain (Server-Cert -> IntermediateCA -> RootCA) while presenting the server certificate to the client for server side authentication. When set to FALSE, Prime Access Registrar sends only the server certificate (Server-Cert) to the client.

**General Properties tab**

The tabs are available if the protocol is set to Radius or Tacacs-and-Radius.

IPAddress	<p>Required; must be a valid IP address and unique in the Clients list. Prime Access Registrar uses this property to identify the Client that sent the request, either using the source IP address to identify the immediate sender or using the <b>NAS-IP-Address</b> attribute in the Request dictionary to identify the NAS sending the request through a proxy.</p> <p>When a range is configured for a Client's IPAddress property, any incoming requests whose source address belongs to the range specified, will be allowed for further processing by the server. Similarly when a wildcard (an asterisk '*' in this case) is specified, any incoming requests whose source address matches the wildcard specification will be allowed. In both the cases, the configured client properties like SharedSecret, and Vendor are used to process the requests.</p> <p>You can specify a range of IP addresses using a hyphen as in:</p> <p style="padding-left: 40px;">100.1.2.11-20</p> <p>You can use an asterisk wildcard to match all numbers in an IP address octet as in:</p> <p style="padding-left: 40px;">100.1.2.*</p> <p>You can specify an IPAddress and a subnet mask together using Classless Inter-Domain Routing (CIDR) notation as in:</p> <p style="padding-left: 40px;">100.1.2.0/24</p> <p>You can use the IPAddress property to set a base address and use the NetMask property to specify the number of clients in the subnet range.</p>
Shared Secret	Required; must match the secret configured in the Client.
Type	Required; accept the default (NAS), or set it to ATM, Proxy, or NAS+Proxy.
Vendor	Optional; you can use this property when you need special processing for a specific vendor's NAS. To use this property, you must configure a <b>Vendor</b> object and include a script. Prime Access Registrar provides five Scripts you can use: one for Ascend, Cisco, Cabletron, Altiga, and one for USR. You can also provide your own Script.

**Table 2-62 Client Properties (continued)**

Fields	Description
NetMask	<p>Specifies the subnet mask used with the network address setting configured for the IPAddress property when configuring a range of IP addresses.</p> <p>This property is not used for a single client with an IP address only. The NetMask property is used to configure multiple clients when you configure a base IP address in the IPAddress property. You can set the NetMask property for a range of 256 clients using the following example:</p> <p style="text-align: center;"><b>set NetMask 255.255.255.0</b></p> <p><b>Note</b> If you set the NetMask property, validation will fail if you attempt to specify a subnet mask using CIDR notation with the IPAddress property (described above).</p>
Enforce Traffic Throttling	By default, the value is set to FALSE. When set to TRUE, the traffic throttling check for the packet will be executed.
<b>Dynamic Authorization tab</b>	
Enable Dynamic Authorization	Optional; when set to TRUE, this property enables Change of Authorization (CoA) and Packet of Disconnect (PoD) features.
Shared Secret	Located under the DynamicAuthorizationServer subdirectory, this is the shared secret used for communicating CoA and PoD packets with the client.
Port	Located under the DynamicAuthorizationServer subdirectory, the default port is 3799.
InitialTimeout	Located under the DynamicAuthorizationServer subdirectory, the default is 5000.
MaxTries	Located under the DynamicAuthorizationServer subdirectory, the default is 3.
COA Attribute	This property is found under the DynamicAuthorizationServer subdirectory and points to a group of attributes to be included in a CoA request sent to this client. These attribute groups are created and configured under the AttributeGroups subdirectory in <b>/Radius/Advanced</b> .
POD Attribute	This property is found under the DynamicAuthorizationServer subdirectory and points to a group of attributes to be included in a POD request sent to this client. These attribute groups are created and configured under the AttributeGroups subdirectory in <b>/Radius/Advanced</b> .
<b>Notification Properties tab</b>	
Enable Notifications	<p>Required; the default value is FALSE and indicates the client is not capable of receiving Accounting-Stop notifications from the Prime Access Registrar server.</p> <p>When set to TRUE, the client can receive Accounting-Stop notifications from the Prime Access Registrar server and additional properties must be configured under a new sub-directory named NotificationProperties.</p>
InitialTimeout	<p>Located under the NotificationProperties subdirectory, specifies the timeout value in milliseconds the Prime Access Registrar server waits for an Accounting-Response packet before attempting a retry (sending another Accounting-Stop packet to the client).</p> <p>Required when EnableNotifications is set to TRUE; the default value is 5000.</p>

**Table 2-62 Client Properties (continued)**

Fields	Description
Port	Located under the NotificationProperties subdirectory, specifies the port used by the Prime Access Registrar server to receive Accounting-Stop packets. Required when EnableNotifications is set to TRUE; the default value is 1813.
MaxTries	Located under the NotificationProperties subdirectory, specifies the number of times the Prime Access Registrar server sends an Accounting-Stop packet to a client.  Required when EnableNotifications is set to TRUE; the default value is 3.
Notification-Properties	When the EnableNotifications property is set to TRUE, this subdirectory contains additional properties required to support the Query-Notify feature.
NotificationAttributeGroup	Located under the NotificationProperties subdirectory, specifies the name of an attribute group under <b>/Radius/Advanced/AttributeGroups</b> that contains the attributes to be included when sending an the Accounting-Stop packet to this client.  Required when EnableNotifications is set to TRUE; there is no default value. You must provide the name of a valid AttributeGroup and the named AttributeGroup must contain at least one valid attribute, or validation will fail.

**TCP Options**

This section is available if the protocol is set to **radius-tls**.

KeepAliveInterval	Time interval in seconds between individual keepalive probes.
TCPConnectionIdleTime	Time (in seconds) the connection can remain idle before TCP starts sending keepalive probes.
KeepAliveMaxTries	Maximum number of keepalive probes TCP can send before dropping the connection.

You can use the Clients page for the following:

- [Filtering Records](#)
- [Adding Client Details](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Client Details

To add new Client details:

- 
- Step 1** Choose **Network Resources > Clients**. The Clients page is displayed.
  - Step 2** Click **Add** to add new client details. The Client Details page is displayed.
  - Step 3** Enter the required details in the General Properties, Dynamic Authorization, and Notification Properties tabs.
  - Step 4** Click **Save** to save the specified details in the Client Details page. Otherwise click **Cancel** to return to the Client page without saving the details.

The Client page is displayed with the newly added details or a respective error message is displayed.

## Remote Servers

You can use the RemoteServers object to specify the properties of the remote servers to which Services proxy requests.

Prime Access Registrar provides the following RemoteServer protocol types:

- [LDAP](#)
- [LDAP Accounting](#)
- [ODBC/OCI](#)
- [ODBC/OCI-Accounting](#)
- [Diameter](#)
- [REST](#)
- [Others](#)



### Note

You must not configure a remote server with an IP address, which is same as that of the client. This is applicable for all types of remote servers.

## LDAP

Specify the **ldap** service type when you want to use a particular LDAP remote server for authentication and/or authorization. When using LDAP for authentication and a local database for authorization, ensure that the usernames in both locations are identical with regard to case-sensitivity.

[Table 2-63](#) lists and describes the fields in the Add LDAP-RemoteServers Details page.

**Table 2-63 LDAP Server Properties**

Fields	Description
<b>LDAP Properties tab</b>	
Name	Required; name of the LDAP server.
Host Name	Required; the LDAP server's hostname or IP address. Prime Access Registrar supports IPv4 and IPv6 addresses for the hostname. <b>Note</b> To use IPv6 addresses, you must have Next Generation (NG) license of Prime Access Registrar. For LDAP, IPv6 addresses must be enclosed in square brackets, as in [2001:420:27c1:420:250:56ff:fe99:3dfd].
Port	Required; defaults to port 389.
Description	Description of the LDAP server.

**Table 2-63** LDAP Server Properties (continued)

Fields	Description
Timeout	<p>Required; the default is 15. The timeout property indicates how many seconds the RADIUS server will wait for a response from the LDAP server.</p> <p><b>Note</b> Use InitialTimeout from above as a template, except this is timeout is specified in seconds.</p>
Reactivate Time Interval	<p>Required; the amount of time (in milliseconds) to wait before retrying a remote server that was offline. You must specify a number greater than zero. The default is 300,000 (5 minutes).</p>
MaxReferrals	<p>Required; must be a number equal to or greater than zero. This property indicates how many referrals are allowed when looking up user information. When you set this property to zero, no referrals are allowed.</p> <p>Cisco Prime Access Registrar manages referrals by allowing the RADIUS server's administrator to indicate an LDAP "referral attribute," which might or might not appear in the user information returned from an LDAP query. When this information is returned from a query, Cisco Prime Access Registrar assumes it is a referral and initiates another query based on the referral. Referrals can also contain referrals.</p> <p><b>Note</b> This is an LDAP v2 referral property.</p>
Referral Attribute	<p>Required when you have specified a <b>MaxReferrals</b> value. This property specifies which LDAP attribute, returned from an LDAP search, to check for referral information.</p> <p><b>Note</b> This is an LDAP v2 referral property.</p>
Referral Filter	<p>Required when you have specified a <b>MaxReferral</b> value. This is the filter Cisco Prime Access Registrar uses when processing referrals. When checking referrals, the information Cisco Prime Access Registrar finds in the referral itself is considered to be the search path and this property provides the filter. The syntax is the same as that of the <b>Filter</b> property.</p> <p><b>Note</b> This is an LDAP v2 referral property.</p>
Bind Name	<p>Optional; the distinguished name (dn) to use when establishing a connection between the LDAP and RADIUS servers.</p>
Bind Password	<p>Optional; the password associated with the <b>BindName</b>.</p>
Search Path	<p>Required; the path that indicates where in the LDAP database to start the search for user information.</p>
Limit Outstanding Requests	<p>Required; the default is FALSE. Cisco Prime Access Registrar uses this property in conjunction with the <b>MaxOutstandingRequests</b> property to tune the RADIUS server's use of the LDAP server.</p> <p>When you set this property to TRUE, the number of outstanding requests for this RemoteServer is limited to the value you specified in <b>MaxOutstandingRequests</b>. When the number of requests exceeds this number, Cisco Prime Access Registrar queues the remaining requests, and sends them as soon as the number of outstanding requests drops to this number.</p>
User Password Attribute	<p>Required; this specifies which LDAP field the RADIUS server should check for the user's password.</p>

**Table 2-63 LDAP Server Properties (continued)**

Fields	Description
Escape Spl.Character in UserName	FALSE by default
Datasource Connections	Specifies the number of concurrent connections to the LDAP server. The default value is 8.
Use SSL	A boolean field indicating whether you want Cisco Prime Access Registrar to use SSL (Secure Socket Layer) when communicating with this RemoteServer. When you set it to TRUE, be sure to specify the <b>CertificateDBPath</b> field in the <b>Advanced</b> section, and be sure the port you specified for this RemoteServer is the SSL port used by the LDAP server.
EnableKeepAlive	Default is FALSE. This is enabled to send a TCP keepalive to keep the idle connection active.
Filter	Required; this specifies the search filter Cisco Prime Access Registrar uses when querying the LDAP server for user information. When you configure this property, use the notation “%s” to indicate where the user ID should be inserted. For example, a typical value for this property is “(uid=%s),” which means that when querying for information about user joe, use the filter uid=joe.
Max Outstanding Requests	Required when you have set the <b>LimitOutstandingRequests</b> to TRUE. The number you specify, which must be greater than zero, determines the maximum number of outstanding requests allowed for this remote server.
Password Encryption Style	The default is <b>None</b> . You can also specify <b>crypt</b> , <b>dynamic</b> , <b>SHA-1</b> , and <b>SSHA-1</b> .
DNSLookup and LDAP RebindInterval	Specifies the timeout period after which the Prime Access Registrar server will attempt to resolve the LDAP hostname to IP address (DNS resolution); 0 by default
Search Scope	Specifies how deep to search within a search path; default is <i>SubTree</i> which indicates a search of the base object and the entire subtree of which the base object distinguished name is the highest object.  <i>Base</i> indicates a search of the base object only.  <i>OneLevel</i> indicates a search of objects immediately subordinate to the base object, but does not include the base object.
Use Binary Password Comparison	A boolean field that enables binary password comparison for authentication. This property when set to TRUE, enables binary password comparison. By default, this property is set to FALSE.
Use Bind Based Authentication	A boolean field that enables bind-based authentication with LDAP server. By default, this property is set to FALSE. When set to FALSE, it uses existing legacy authentication method.  On setting this property to TRUE, the mappings LDAPToRadius, LDAP-ToEnvironment, and LDAPToCheckItem will not work.
<b>LDAPToRadiusMappings tab</b>	
LDAPAttribute	Set the value for the LDAP attribute

Table 2-63 LDAP Server Properties (continued)

Fields	Description
RadiusAttribute	<p>A list of name/value pairs in which the name is the name of the <b>ldap</b> attribute to retrieve from the user record, and the value is the name of the RADIUS attribute to set to the value of the <b>ldap</b> attribute retrieved.</p> <p>For example, when the <b>LDAPToRadiusMappings</b> has the entry: <b>FramedIPAddress = Framed-IP-Address</b>, the RemoteServer retrieves the <b>FramedIPAddress</b> attribute from the <b>ldap</b> user entry for the specified user, uses the value returned, and sets the Response variable <b>Framed-IP-Address</b> to that value.</p> <p>Click the <b>Add</b> button to save the details and list it in the attribute list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.</p>
<b>LDAPToCheckItems Mappings tab</b>	
Attribute Type	Select either <b>RADIUS</b> or <b>VENDOR</b> . If Vendor is selected, specify the vendor type from the drop-down list.
LDAPAttribute	Set the value for the LDAP attribute
CheckedItems	<p>A list of LDAP <i>attribute/value</i> pairs which must be present in the RADIUS access request and must match, both name and value, for the check to pass.</p> <p>For example, when the <b>LDAPToCheckItemMappings</b> has the entry: <b>group = User-Group</b>, the Access Request must contain the attribute <b>group</b>, and it must be set to <b>User-Group</b>.</p> <p>Click the <b>Add</b> button to save the details and list it in the attribute list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.</p>
<b>LDAPToEnvironmentalMappings tab</b>	
LDAPAttribute	Set the value for the LDAP attribute
EnvironmentalAttribute	<p>A list of name/value pairs in which the name is the name of the <b>ldap</b> attribute to retrieve from the user record, and the value is the name of the Environment variable to set to the value of the <b>ldap</b> attribute retrieved.</p> <p>For example, when the <b>LDAPToEnvironmentMappings</b> has the entry: <b>group = User-Group</b>, the RemoteServer retrieves the <b>group</b> attribute from the <b>ldap</b> user entry for the specified user, uses the value returned, and sets the Environment variable <b>User-Group</b> to that value.</p> <p>Click the <b>Add</b> button to save the details and list it in the attribute list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.</p>

You can use the LDAP-RemoteServers page for the following:

- [Filtering Records](#)
- [Adding LDAP Details](#)
- [LDAP Accounting](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding LDAP Details

To add new LDAP details:

- 
- Step 1** Choose **Network Resources > RemoteServers > LDAP**. The LDAP-RemoteServers page is displayed.
- Step 2** Click **Add** to add LDAP details. The LDAP-RemoteServers Details page is displayed.
- Step 3** Enter the required details in the tabs.
- Step 4** Click **Save LDAP Server** to save the specified details in the LDAP-RemoteServers Details page. The LDAP-RemoteServers page is displayed with the newly added details or a respective error message is displayed. Otherwise click **Cancel** to return to the LDAP-RemoteServers page without saving the details.
- 

## LDAP Accounting

Previous releases of Prime Access Registrar supported accessing user data from an LDAP server, but this feature was limited to performing authentication and authorization (AA). You could only write the accounting records to local file or oracle database or proxy to another RADIUS server. Prime Access Registrar supports writing accounting records into LDAP server enabling integration between billing systems and LDAP.

[Table 2-64](#) lists and describes the fields in the LDAPAcct RemoteServer Details page.

**Table 2-64 LDAP Accounting Server Properties**

Fields	Description
<b>LDAP Acct Properties tab</b>	
Name	Name of the remote server; this property is mandatory, and there is no default.
Description	Optional description of server.
HostName	Required; the LDAP server's hostname or IP address.
Port	Required; the default value is 389. Port the LDAP server is listening on.
Timeout	Mandatory time interval (in seconds) to wait for LADP-write operation to complete; defaults to 15 seconds.
ReactivateTimerInterval	Mandatory time interval (in milliseconds) to activate an inactive server; defaults to 300000 ms.

**Table 2-64 LDAP Accounting Server Properties (continued)**

<b>Fields</b>	<b>Description</b>
BindName	Optional; the distinguished name (dn) to use when establishing a connection between the LDAP and RADIUS servers.
BindPassword	Optional; the password associated with the BindName.
EnableKeepAlive	Required; default is FALSE. This is enabled to send a TCP keepalive to keep the idle connection active.
Delimiter	Character used to separate the values of the attributes given in Attribute-List property.
LDAPEnvironmentMultiValueDelimiter	Optional; allows you to specify a character that separates multi-valued attribute lists when using ldap-accounting.
DnPath	Required; the path that indicates where in the LDAP database to start the write for user information.
EntryName	Required; this specifies the write entry name Prime Access Registrar uses when inserting the LDAP server for user information. When you configure this property, use the notation "%s" to indicate where the user ID should be inserted. For example, a typical value for this property is "(uid=%s)," which means that when inserting for information about user joe, use the entry name uid=joe.
LimitOutstandingRequests	Required; the default is FALSE. Prime Access Registrar uses this property in conjunction with the <b>MaxOutstandingRequests</b> property to tune the RADIUS server's use of the LDAP server.  When you set this property to TRUE, the number of outstanding requests for this RemoteServer is limited to the value you specified in <b>MaxOutstandingRequests</b> . When the number of requests exceeds this number, Prime Access Registrar queues the remaining requests, and sends them as soon as the number of outstanding requests drops to this number.
MaxOutstandingRequests	Required when you have set the <b>LimitOutstandingRequests</b> to TRUE. The number you specify, which must be greater than zero, determines the maximum number of outstanding requests allowed for this remote server.
ObjectClass	Required; list of object classes which are all schemas defined in LDAP server. These schemas define required attributes and allowed attributes for an entry which is inserted from Prime Access Registrar.
DNSLookup and LDAPAcct RebindInterval	Specifies the timeout period after which the Prime Access Registrar server will attempt to resolve the LDAP hostname to IP address (DNS resolution).
Escape Spl.Character in UserName	FALSE by default.
AttributeList	List of comma-separated attribute names.
Datasource Connections	Mandatory number of connections to be established; defaults to 8.
UseLocalTimeZone	Optional; the default is FALSE. It determines the timezone of accounting records TimeStamp.

Table 2-64 LDAP Accounting Server Properties (continued)

Fields	Description
UseSSL	A boolean field indicating whether you want Prime Access Registrar to use SSL (Secure Socket Layer) when communicating with this RemoteServer. When you set it to TRUE, be sure to specify the <b>CertificateDB-Path</b> field in the <b>Advanced</b> section, and be sure the port you specified for this RemoteServer is the SSL port used by the LDAP server.
<b>AttributestoWrite tab</b>	
LDAPAcctAttribute	Set the LDAP Accounting attribute.
EnvironmentalAttribute	A list of name and value pairs in which the name is the name of the data store attribute to retrieve from the user record, and the value is the name of the RADIUS attribute to set to the value of the data store attribute retrieved. The data store attributes must match those defined in the external SQL file.  Click the <b>Add</b> button to save the details and list it in the Attributes list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.

You can use the LDAP Acct-RemoteServers page for the following:

- [Filtering Records](#)
- [Adding LDAP Accounting Details](#)
- [Editing Records](#)
- [Deleting Records](#)

### Adding LDAP Accounting Details

To add new LDAP accounting details:

- 
- Step 1** Choose **Network Resources > RemoteServers > LDAP Accounting**. The LDAPAcct-RemoteServers page is displayed.
  - Step 2** Click **Add** to add LDAP accounting details. The LDAPAcct RemoteServer Details page is displayed.
  - Step 3** Enter the required details in the tabs.
  - Step 4** Click **Save LDAP Acct Server** to save the specified details in the LDAPAcct RemoteServer Details page. Otherwise click **Cancel** to return to the LDAPAcct-RemoteServers page without saving the details. The LDAPAcct-RemoteServers page is displayed with the newly added details or a respective error message is displayed.
-

## ODBC/OCI

Specify **odbc** or **oci** when you want to use an ODBC or OCI service for authentication, authorization and accounting through an ODBC or OCI data store respectively. Use an ODBC or OCI service to authenticate and authorize an access requests by querying user information through ODBC or OCI and to insert accounting records into a data store through ODBC or OCI.


**Note**

The ODBC service supports MYSQL database service and OCI supports Oracle with 10.2.0 to 11.2.0 Oracle client.

Table 2-65 lists and describes the fields in the ODBC/OCI-RemoteServers Details page.

**Table 2-65 ODBC/OCI Server Properties**

Fields	Description
Name	Required; name of the ODBC/OCI Server.
Protocol	The type of remote server. You select the option ODBC or OCI from the drop-down list.
Datasource Connections	Required; default is 8. This represents the total number of connections Prime Access Registrar can open with the ODBC server; total number of threads Prime Access Registrar can create for the ODBC server.
ODBC Datasource Name	Required; name of the ODBCDataSource to use and must refer to one entry in the list of ODBC datasources configured under <b>/Radius/Advanced/ODBCDataSources</b> .
User Password Attribute	Set the user password.
SNMPTrapIP	The SNMP trap IP for the remote servers. Prime Access Registrar supports IPv4 and IPv6 addresses for the SNMP trap IP. <b>Note</b> To use IPv6 addresses, you must have Next Generation (NG) license of Prime Access Registrar.
Description	Description of the ODBC Server
Timeout	Required; the default is 15. The timeout property indicates how many seconds the RADIUS server will wait for a response from the ODBC server. <b>Note</b> Use InitialTimeout from above as a template, except this is timeout is specified in seconds.
Reactivate Time Interval	Required; default is 300,000 milliseconds. Length of time to wait before attempting to reconnect if a thread is not connected to a data source.
Keep Alive Timer Interval	Mandatory time interval (in milliseconds) to send a keepalive to keep the idle connection active; defaults to zero (0) meaning the option is disabled
SNMPTrapPort	The SNMP trap port for the remote server; defaults to 1521.

Table 2-65 ODBC/OCI Server Properties (continued)

Fields	Description
OCITimeOutCount	This and the following fields appear when you select <b>oci</b> from the <b>Protocol</b> drop-down list.  Required; continuous timeout count to disconnect the selected connection. Default value is 10.
OCIConnectionReactivationInterval	Required; time interval for attempting to reconnect the disconnected OCI remote server session. Default value is 3000 ms.
OCIActiveConnectionThresholdCount	Required; threshold count of disconnections after which Prime Access Registrar will mark the remote server as down and try to reactivate it. Default value is 4.
<b>SQL Definitions tab</b>	
Name	SQLDefinition properties define the SQL you want to execute.
Description	Description of the SQL
Type	Prime Access Registrar supports only type <b>query</b> .
SQL	SQL query used to add, update or delete a record from a database
Execution SequenceNumber	Sequence number for SQLStatement execution, must be greater than zero (mandatory, no default)
Marker List	Defines all markers for the query. MarkerList uses the format UserName/SQL_DATA_TYPE.
<b>RadiusMappings tab</b>	
ODBC/OCI Attribute	Set the ODBC or OCI attribute
RADIUS Attribute	A list of name and value pairs in which the name is the name of the data store attribute to retrieve from the user record, and the value is the name of the RADIUS attribute to set to the value of the data store attribute retrieved. The data store attributes must match those defined in the external SQL file.  Click the <b>Add</b> button to save the details and list it in the Attributes list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.
<b>CheckItemsMappings tab</b>	
Attribute Type	Select either <b>RADIUS</b> or <b>VENDOR</b> . If Vendor is selected, specify the vendor type from the drop-down list.
ODBC/OCI Attribute	Set the ODBC or OCI attribute
CheckItem	A list of ODBC attribute/value pairs.  Click the <b>Add</b> button to save the details and list it in the Attributes list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.
<b>EnvironmentalMappings tab</b>	

**Table 2-65 ODBC/OCI Server Properties (continued)**

Fields	Description
ODBC/OCI Attribute	Set the ODBC or OCI attribute
Environmental Attribute	<p>A list of name/value pairs in which the name is the name of the data store attribute to retrieve from the user record, and the value is the name of the Environment variable to set to the value of the ODBC attribute retrieved.</p> <p>Click the <b>Add</b> button to save the details and list it in the Attributes list. To navigate between the listed attributes, use the navigation option available adjacent to the list. See <a href="#">Relocating Records</a> for more details. To delete the available attributes, select the relevant attribute and click the <b>Delete</b> button below.</p>

You can use the ODBC/OCI-RemoteServers page for the following:

- [Filtering Records](#)
- [Adding ODBC/OCI Details](#)
- [ODBC/OCI-Accounting](#)
- [Editing Records](#)
- [Deleting Records](#)

### Adding ODBC/OCI Details

To add new ODBC or OCI details:

- 
- Step 1** Choose **Network Resources > RemoteServers > ODBC/OCI**. The ODBC/OCI-RemoteServers page is displayed.
  - Step 2** Click **Add** to add ODBC or OCI details. The ODBC/OCI-RemoteServers Details page is displayed.
  - Step 3** Enter the required details.
  - Step 4** Click **Add** to enter the SQL details in the **SQL Definitions** tab.
  - Step 5** Click **Save** to save the specified details in the **SQL Definitions** tab or click **Cancel** to cancel the action.
  - Step 6** Enter the required details in the tabs.
  - Step 7** Click **Add Server** to save the specified details in the ODBC/OCI-RemoteServers Details page. Otherwise click **Cancel** to return to the ODBC/OCI-RemoteServers page without saving the details.
- The ODBC/OCI-RemoteServers page is displayed with the newly added details or a respective error message is displayed.
- 

### ODBC/OCI-Accounting

If you use the Oracle Accounting feature, you must configure an ODBC/OCI-Accounting RemoteServer object.

Table 2-66 lists and describes the fields in the Add ODBC/OCI Accounting-RemoteServers page.

**Table 2-66 ODBC/OCI Accounting Server Properties**

Fields	Description
<b>General Properties tab</b>	
Name	Name of the remote server; this property is mandatory, and there is no default.
Protocol	The type of Accounting remote server. You can select the option <code>odbc-accounting</code> or <code>oci-accounting</code> from the drop-down list.
Datasource Connections	Mandatory number of connections to be established; defaults to 8
ODBC Datasource Name	Name of the ODBCDataSource to use and must refer to one entry in the list of ODBC datasources configured under <b>/Radius/Advanced/ODBCDataSources</b> . Mandatory; no default
Buffer Accounting Packets	Mandatory, TRUE or FALSE, determines whether to buffer the accounting packets to local file, defaults to TRUE which means that packet buffering is enabled.  <b>Note</b> When set to TRUE, a constant flow of incoming accounting packets can fill the buffer backing store files in <b>/cisco-ar/data/odbc</b> beyond the size configured in <code>MaximumBufferFileSize</code> . Configure <code>BackingStoreDiscThreshold</code> in <b>/Radius/Advanced</b> when using ODBC accounting.
Max. Buffer Filesize	Mandatory if <code>BufferAccountingPackets</code> is set to TRUE, determines the maximum buffer file size, defaults to 10 Megabyte)
Backing Store Environment Variables	Optional; when <code>BufferAccountingPackets</code> is set to TRUE, contains a comma-separated list of environment variable names to be stored into a local file along with buffered packet. No default. <code>BackingStoreEnvironmentVariables</code> can also be specified in scripts using the <code>BackingStoreEnvironmentVariables</code> environment variable.
Attribute List	List of comma-separated attribute names.
SNMPTrapIP	Optional; when set to a valid IP address, the traps (responding/not responding traps) for the ODBC/OCI Accounting server will have this IP address. This is used to identify the server. If the value is not set, SNMP traps use 255.255.255.255 as the IP address.
Description	Optional; description of server.
Timeout	Mandatory time interval (in seconds) to wait for SQL operation to complete; defaults to 15 seconds.
Reactivate Time Interval	Mandatory time interval (in milliseconds) to activate an inactive server; defaults to 300000 ms.
Keep Alive Timer Interval	Mandatory time interval (in milliseconds) to send a keepalive to keep the idle connection active; defaults to zero (0) meaning the option is disabled.
No. of Retries for Buffered Packet	Mandatory if <code>BufferAccountingPackets</code> is set to TRUE. A number greater than zero determines the number of attempts to be made to insert the buffered packet into Oracle. Defaults to 3.

**Table 2-66 ODBC/OCI Accounting Server Properties (continued)**

Fields	Description
Use Local Timezone	Set to TRUE or FALSE, determines the timezone of accounting records' TimeStamp (defaults to FALSE).
Delimiter	Character used to separate the values of the attributes given in AttributeList property.
SNMPTrapPort	Optional; when set to a valid port, the traps (responding/not responding traps) for the ODBC/OCI Accounting server will have this port. If the value is not set, SNMP traps use 1521 as the IP port.
OCIAutoCommit	This and the following fields appear when you select <b>oci-accounting</b> from the <b>Protocol</b> drop-down list.  Required; check this box to let the OCI remote server connections auto commit the Oracle database transactions. Prime Access Registrar will not execute the Commit query explicitly to commit the Oracle database transactions. Auto Commit flag is enabled while establishing the connection itself.
OCITransactionCount	Required; default value is zero. Number of transactions per connection after which Prime Access Registrar can execute the Commit query in the Oracle database instead of committing the transactions after each Oracle query.
OCITimeOutCount	Required; continuous timeout count to disconnect the selected connection. Default value is 10.
OCIConnectionReactivationInterval	Required; time interval for attempting to reconnect the disconnected OCI remote server session. Default value is 3000 ms.
OCIActiveConnectionThresholdCount	Required; threshold count of disconnections after which Prime Access Registrar will mark the remote server as down and try to reactivate it. Default value is 4.
<b>SQL Definitions tab</b>	
Name	Required; SQLDefinition properties define the SQL you want to execute.
Description	Description of the SQL
Type	Required; Prime Access Registrar supports insert, update and delete options.
SQL	Required; SQL query used to acquire the password
Execution SequenceNumber	Required; sequence number for SQLStatement execution, must be greater than zero (mandatory, no default)
Marker List	Required; defines all markers for the query. MarkerList uses the format UserName/SQL_DATA_TYPE.

You can use the ODBC/OCI Accounting-RemoteServers page for the following:

- [Filtering Records](#)
- [Adding ODBC/OCI Accounting Details](#)
- [Others](#)
- [Editing Records](#)

- [Deleting Records](#)

## Adding ODBC/OCI Accounting Details

To add new ODBC or OCI accounting details:

- 
- Step 1** Choose **Network Resources > RemoteServers > ODBC/OCI Accounting**. The ODBC/OCI Accounting-RemoteServers page is displayed.
- Step 2** Click **Add** to add ODBC or OCI accounting details. The ODBC/OCI Accounting-RemoteServers Details page is displayed.
- Step 3** Enter the required details in the tabs.
- Step 4** Click **Add Accounting Server** to save the specified details in the ODBC/OCI Accounting-RemoteServers Details page. The ODBC/OCI Accounting-RemoteServers page is displayed with the newly added details or a respective error message is displayed. Otherwise click **Cancel** to return to the ODBC/OCI Accounting-RemoteServers page without saving the details.
- 

## Diameter

Diameter is a networking protocol which is derived from RADIUS protocol.

You can click the **Add** button in the Diameter-RemoteServers page to add a new Diameter remote server. [Table 2-67](#) lists and describes the Diameter remote server properties.

**Table 2-67** *Diameter Remote Server Properties*

Fields	Description
Name	Required; name of the Diameter server.
Description	Optional; description of the Diameter server.
Protocol	Required; protocol used by the Diameter server.
MaxTries	Number of retry attempts to be made by the Diameter server for request and response.
Host Name	Host name of the server.
Initial Timeout	Specifies the timeout value in milliseconds the Prime Access Registrar server waits for an Accounting-Response packet before attempting a retry.  This value must be less than the DWatchDogTimeout value.
DestinationPort	Port used by the server.
DWatchDogTimeout	Time interval between watch dog messages.
IncomingScript	Optional; if there is a script, it is the first script Prime Access Registrar runs when it receives a request from any client and/or for any service.

**Table 2-67** *Diameter Remote Server Properties (continued)*

Fields	Description
OutgoingScript	Optional; if there is a script, it is the last script Prime Access Registrar runs before it sends a Diameter packet to the remote server.  You can choose to configure blacklisting as part of the outgoing script for Diameter remote server. For more information about blacklisting, see the “Using Extension Points” chapter of the <a href="#">Cisco Prime Access Registrar 8.0 Administrator Guide</a> .
SCTP-Enabled	Indicates whether the connection will be an SCTP. If set to TRUE, SCTP will be used. If set to FALSE, TCP will be used.
AdvertiseHostName	Optional; specifies the local hostname address that will be advertised by the Prime Access Registrar server to other peers during CER/CEA exchange.
AdvertiseRealm	Advertising realm.
ReactivateTimerInterval	Mandatory time interval, in milliseconds, to reactivate an inactive server.
Vendor	Select a valid vendor.
LimitOutstandingRequests	Check this box to limit the number of outstanding requests. If you enable this option, the number of outstanding requests for the Diameter remote server is limited to the value specified in the MaxOutstandingRequests field.
UserLogEnabled	Check this box to log user details of the specified remote server. If this option is enabled, Prime Access Registrar stores all subscriber messages including Diameter request and response in a separate log file called subscriber_log in the \$INSTALLPATH/logs folder.
MaxOutstandingRequests	Maximum number of outstanding requests allowed for the Diameter remote server
MaxPendingPackets	Maximum number of packets that can be pending for the Diameter remote server.
DestinationRealm	Required. Destination realm to send Diameter packets to the remote server. The role of the remote server should be Relay.
TLS-Enabled	Check this box to enable TLS security mechanism for the Diameter remote server.
MaxTPSLimit	Maximum number of requests allowed per second for the Diameter remote server.
MaxSessionLimit	Maximum number of sessions allowed for the Diameter remote server.
DisconnectBasedOn-Threshold	Check this box if the remote server’s TCP connections are to be disconnected based on a threshold value.
DisconnectThreshold	This field appears only when the DisconnectBasedOnThreshold box is checked.  Threshold count to disconnect the remote server’s TCP connections, which indicates the total number of failed requests that are not answered even after MaxTries is reached for each of those requests.
Host	Destination host to send the packets (default is localhost).

**Table 2-67 Diameter Remote Server Properties (continued)**

Fields	Description
<b>SCTPParameters Section</b>	
This section is available if the SCTP-Enabled option is checked.	
SourcePort	Remote server source port. Default value is 3868.
DestinationPort	Remote server destination port. Default value is 3868.
PathMaxRetrans	Maximum number of consecutive retransmissions over a destination transport address of a peer endpoint before it is marked as inactive. Default value is 5.
RTOInitial	Initial value of RTO (retransmission timeout) that is used in RTO calculations. Measured in milliseconds and default value is 3 seconds.
RTOCookieLife	Maximum lifespan of the cookie sent in an INIT ACK chunk. Measured in milliseconds and default value is 60 secs.
RTOMin	Minimum value of RTO. Measured in milliseconds and default value is 1 second.
HBInterval	Interval when a HEARTBEAT chunk is sent to a destination transport address to monitor the reachability of an idle destination transport address. Measured in milliseconds and default is 30 seconds.
RTOMax	Maximum value of RTO. Measured in milliseconds and default value is 60 seconds.
SACKTimeout	Delayed SACK timeout. Default value is 200 msec.
MaxInitRetransmits	Maximum number of times an INIT chunk or a COOKIE ECHO chunk is retransmitted before an endpoint aborts the initialization process and closes the association. Default value is 8.
InitNumOStreams	Initial number of streams per socket.
AssociationMaxRetrans	Maximum number of consecutive retransmissions to a peer before an endpoint considers that the peer is unreachable and closes the association. Default value is 10.
InitMaxInStreams	Maximum number of inbound streams per socket.
<b>SCTPAdvHostName Section</b>	
This section is available if the SCTP-Enabled option is checked.	
Local	SCTP advertising host name of the local server.
Remote	SCTP advertising host name of the remote server.
<b>TLSEnabled Section</b>	
This section is available if the TLS-Enabled option is checked.	
PrivateKeyPassword	The password used to protect the server's private key.

Table 2-67 Diameter Remote Server Properties (continued)

Fields	Description
ServerKeyFile	<p>The full pathname of the file containing the server's RSA private key. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The valid encoding prefix is "PEM". If an encoding prefix is not present, the file is assumed to be in PEM format.</p> <p>The following example assumes that the subdirectory <b>pki</b> under <b>/cisco-ar</b> contains the server's certificate file. The file <b>server-key.pem</b> is assumed to be in PEM format. The file extension <b>.pem</b> is not significant.</p> <p style="text-align: center;"><b>set ServerKeyFile PEM:/cisco-ar/pki/server-key.pem</b></p>
ServerCertificateFile	<p>The full pathname of the file containing the server's certificate or certificate chain used during the TLS exchange. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The valid encoding prefix is PEM. If an encoding prefix is not present, the file is assumed to be in PEM format.</p>
CACertificateFile	<p>The full pathname of the file containing trusted CA certificates used for client verification. The file can contain more than one certificate, but all certificates must be in PEM format. DER encoding is not allowed.</p>
CACertificatePath	<p>The name of a directory containing trusted CA certificates (in PEM format) used for client verification. This parameter is optional, and if it is used there are some special preparations required for the directory it references.</p> <p>Each certificate file in this directory must contain exactly one certificate in PEM format. The server looks up the certificate files using the MD5 hash value of the certificate's subject name as a key. The directory must therefore also contain a set of symbolic links each of which points to an actual certificate file. The name of each symbolic link is the hash of the subject name of the certificate.</p> <p>For example, if a certificate file named <b>ca-cert.pem</b> is located in the CACertificatePath directory, and the MD5 hash of the subject name contained in <b>ca-cert.path.pem</b> is 1b96dd93, then a symbolic link named 1b96dd93 must point to <b>ca-cert.pem</b>.</p> <p>If there are subject name collisions such as multiple certificates with the same subject name, each link name must be indexed with a numeric extension as in 1b96dd93.0 and 1b96dd93.1.</p>
PeerVerificationMode	<p>Select one of the following options:</p> <ul style="list-style-type: none"> <li>• None—If Prime Access Registrar is not required to provide its certificate; but, the peer's certificate must be verified.</li> <li>• Optional—If Prime Access Registrar can provide its certificate optionally; but, the peer's certificate must be verified.</li> <li>• RequireCertificate—If Prime Access Registrar must provide its certificate and the peer's certificate must also be verified.</li> </ul>

**Table 2-67** Diameter Remote Server Properties (continued)

Fields	Description
VerificationDepth	Specifies the maximum length of the certificate chain used for client verification.
EnableAutoChaining	When set to TRUE, Prime Access Registrar sends its server certificate chain (Server-Cert -> IntermediateCA -> RootCA) while presenting the server certificate to the client for server side authentication. When set to FALSE, Prime Access Registrar sends only the server certificate (Server-Cert) to the client.

You can use the Domain Authentication-RemoteServers page for the following:

- [Filtering Records](#)
- [ODBC/OCI](#)
- [Editing Records](#)
- [Deleting Records](#)

### Adding Diameter Remote Server Details

To add new Diameter remote server details:

- 
- Step 1** Choose **Network Resources > RemoteServers > Diameter**. The Diameter-Remote Servers page is displayed.
  - Step 2** Click **Add** to add Diameter remote server details.
  - Step 3** Enter the required details as described in [Table 2-67](#).
  - Step 4** Click **Add Diameter Server** to save the details. Click **Cancel** to return to the previous page without saving the details.

The Diameter-Remote Servers page is displayed with the newly added details or a respective error message is displayed.

---

## REST

Prime Access Registrar allows you to configure a REST remote server for extended-EAP service. Extended-EAP is used as an authorization service to retrieve authorization information from the remote web server using the REST interface. Prime Access Registrar processes all EAP requests and extends through extended EAP service. Extended-EAP is supported for the following EAP protocols:

- EAP-AKA
- EAP-AKA-PRIME
- EAP-SIM

You can click the **Add** button in the **REST-RemoteServers** page to add a new REST remote server. [Table 2-68](#) lists and describes the REST remote server properties.

**Table 2-68 REST Remote Server Properties**

Fields	Description
<b>RESTRemoteServerProperties Tab</b>	
Name	Required; name of the REST server.
Description	Optional; description of the REST server.
Protocol	Indicates the protocol, which is REST.
ReactivateTimerInterval	Required; time interval, in milliseconds, to reactivate an inactive REST server. Default value is 300000.
Timeout	Required; timeout value, in milliseconds, the REST server can wait for a request or response before attempting a retry. Default value is 15. We recommend that you set the value to 1000.
MaxTimeOuts	Maximum number of timeouts allowed for the remote server.
RESTSourceConnections	Mandatory number of connections to be established towards the REST server; default value is eight.
RequestURL	Required; URL of the REST web server including port number. Ensure that you enter IMSI keyword in the URL.
UserName	Required; user name of the REST web server.
Password	Required; password of the REST web server.
KeepAliveTimerInterval	Mandatory time interval, in milliseconds, to send a keepalive to keep the idle connection active; defaults to zero (0) meaning the option is disabled.
<b>RequestToJSONRequestMappings Tab</b>	
RESTAttribute	REST attribute
JSONAttribute	JSON attribute to map to the REST attribute.

You can use the REST RemoteServer page for the following:

- [Filtering Records](#)
- [Editing Records](#)
- [Deleting Records](#)

### Adding REST Remote Server Details

To add new REST remote server details:

- 
- Step 1** Choose **Network Resources > RemoteServers > REST**. The **REST-RemoteServers** page is displayed.
  - Step 2** Click **Add** to add REST remote server details.
  - Step 3** Enter the required details as described in [Table 2-68](#).
  - Step 4** Click **Save REST Server** to save the details. Click **Cancel** to return to the previous page without saving the details.

The REST-RemoteServers page is displayed with the newly added details or a respective error message is displayed.

## Others

This feature of GUI allows you to set other specifications. The various types of protocols are:

- Radius
- Dynamic DNS
- Map-Gateway
- Prepaid-CRB
- Prepaid IS 835C
- Sigtran
- Sigtran-m3ua

[Table 2-69](#) lists and describes the fields in the Remote Server Details page. The fields listed below are the entire list of all the available protocols. The fields are displayed based on the type of protocol selected.

**Table 2-69 Other Server Properties**

Fields	Description
<b>Remote Server Details</b>	
Name	Required; name of the server.
Description	Optional; description of the server.
Protocol	Required; type of the remote server. Choose from one of the following options: <ul style="list-style-type: none"> <li>• Radius</li> <li>• Dynamic DNS</li> <li>• Map-Gateway</li> <li>• Prepaid-CRB</li> <li>• Prepa-IS835C</li> <li>• Sigtran</li> <li>• Sigtran-m3ua</li> </ul>
IP Address	Required; this property specifies where to send the proxy request. It is the address of the remote server. You must set it to a valid IP address.
Port	By default, Prime Access Registrar listens on ports 1812 and 1813.
ReactivateTimerInterval	Mandatory time interval (in milliseconds) to activate an inactive server; defaults to 300000 ms.
MaxTries	Number of times the server tries to send dynamic updates to a server.

**Table 2-69 Other Server Properties (continued)**

<b>Fields</b>	<b>Description</b>
Initial Timeout	Time, in milliseconds, that the server waits for a response before retrying a request.
SharedSecret	Required; the secret shared between the remote server and the RADIUS server.
Vendor	Optional; when set, must be the name of a known Vendor.
IncomingScript	Optional; when set, must be the name of a known incoming script. Prime Access Registrar runs the IncomingScript after it receives the response.
OutGoingScript	Optional; when set, must be the name of a known outgoing script. Prime Access Registrar runs the OutgoingScript just before it sends the proxy request to the remote server.
AccountingPort	Port where the RADIUS server sends accounting packets.
AcknowledgeAccounting	When ACKAccounting is TRUE, the Prime Access Registrar server waits for the Accounting-Response from the remote RADIUS server before sending the corresponding Accounting-Response to the client.  When ACKAccounting is FALSE, the Prime Access Registrar server returns an Accounting-Response to the client without waiting for a response from the remote server.
SendandForget	This field is available if the AcknowledgeAccounting option is disabled.  After forwarding a proxy packet to the remote server and an initial response to the client, Prime Access Registrar maintains a buffer of the original request and a copy of the proxy request until it receives a response from the remote server or packet timeout is triggered.  If SendandForget is enabled, Prime Access Registrar deletes the original and proxy requests from the buffer after sending the response to the client. This helps in reducing buffer pool exhaustion in case of a low-responding remote server.
Accept Dynamic Authorization Requests	The value is set to False, by default.
MaxRename Retries	Number of times that the resource managers can try to add a host even if it detects that the host's name is already present. This controls the number of times Prime Access Registrar tries to modify a host's name to resolve a conflict on each failed update.
MaxTPSLimit	Maximum number of requests allowed per second for the remote server.  This field is available only for RADIUS remote server.
MaxSessionLimit	Maximum number of sessions allowed for the remote server.  This field is available only for RADIUS and Sigtran-m3ua remote servers.

**Table 2-69 Other Server Properties (continued)**

<b>Fields</b>	<b>Description</b>
Trim HostName	Controls whether Prime Access Registrar trims the hostname string to the first period character. If this attribute is enabled, the hostname is truncated before the period. If disabled, the server retains the period characters in the hostname.
FwdZoneTSIG	Server-wide security key to process all forward zone dynamic DNS updates. This is used if a ForwardZoneTSIGKey was not specified on the Resource Manager.
ReverseZoneTSIG	Server-wide security key to process all reverse zone dynamic DNS updates. This is used if a ReverseZoneTSIGKey was not specified on the Resource Manager.
File Name	Name of the shared library provided by the billing server vendor, such as <b>libprepaid.so</b>
Connections	Number of threads the prepaid service and billing server can each use (default is 8).
HostName	Required; hostname of the remote server.
Local Sub System Number	Required; the default value for this property is 0. This represents the subsystem number used by SUA user.
CgPA Global Title Address	Required; represents the Global Title Address of CallingPartyAddress.
Set OPC In CgPA	Required; if it is set to TRUE, OPC will be used in CallingPartyAddress.
CdPAnumberingPlan	Required; used to specify the numbering plan of the called party. The default value is 7.
CgPAnumberingPlan	Required; used to specify the numbering plan of the calling party. The default value is 7.
Global Title Translation Script	This is used to specify the name of the script which is responsible for translating IMSI to GTA.  You can choose to configure blacklisting as part of the global title translation script for SIGTRAN-M3UA remote server. For more information about blacklisting, see the “Using Extension Points” chapter of the <a href="#">Cisco Prime Access Registrar 8.0 Administrator Guide</a> .
SUA Configuration Filename	Required; used to specify the name of configuration file for SUA stack initialization.
Max Outstanding Requests	This represents the maximum outstanding request to HLR.
Timeout	Required; represents the how long the remote server should wait before marking the request as timedout.
Limit Outstanding Requests	Limits the outstanding request to HLR when it is set to TRUE.
SourceIPAddress	Required; name of the local IP address.
SourcePort	Required; specify the port number in which Prime Access Registrar is installed for M3UA transactions.
LocalSubSystemNumber	Required; the local sub system number is set as 149 by default.

**Table 2-69 Other Server Properties (continued)**

<b>Fields</b>	<b>Description</b>
DestinationPort	Required; specify the destination port number to which Prime Access Registrar connects.
IMSITranslationScript	Specify the scripting point that is used to modify the IMSI based on the requirement before sending the request to STP/HLR.
Timeout	Required; specify the time (in seconds) to wait before an authentication request times out; defaults to 120.
ReactivateTimerInterval	Required; specify the time interval (in milliseconds) to activate an inactive server; defaults to 300000 ms (which is 5 minutes).
Limit Outstanding Requests	Prime Access Registrar uses this property in conjunction with the MaxOutstandingRequests property to tune the RADIUS server's use of the HLR. The default is FALSE.  When you set this property to TRUE, the number of outstanding requests for this RemoteServer is limited to the value you specified in MaxOutstandingRequests. When the number of requests exceeds this number, Prime Access Registrar queues the remaining requests, and sends them as soon as the number of outstanding requests drops to this number.
MaxOutstandingRequests	Required; specify the maximum number of outstanding requests allowed for this remote server.
MAP-Version	Required; specify the MAP version as 2 or 3 that HLR supports.
NetworkVariant	Required for SIGTRAN-M3UA remote server; Choose <b>ITU</b> or <b>ANSI</b> to represent the standard that SIGTRAN-M3UA remote server supports.
SubServiceField	Required; specify the type of network to which this SAP belongs. The possible options are INT and NAT which represents international network and national network respectively.
TCAPVariant	Required; specify the name of the TCAP network variant switch. The possible options are ITU88, ITU92, or ITU96.
NetworkAppearance	Required; specify the network appearance code which ranges from 0-2147483647.  This field is optional for SIGTRAN-M3UA remote servers as per the RFC 4666 ( <a href="http://tools.ietf.org/html/rfc4666">http://tools.ietf.org/html/rfc4666</a> .) You can set the value to 0 to remove network appearance from the data packet.
NetworkIndicator	Required; specify the network indicator used in SCCP address. The possible options are NAT and INT which represents international network and national network respectively.
RoutingIndicator	Required; specify the routing indicator. The possible options are RTE_GT or RTE_SSN which is used to route the packets for HLR.

**Table 2-69 Other Server Properties (continued)**

Fields	Description
MLCNumber	<p>Required; specify the MLC number which is required for M3UA service for fetching the MSISDN from the HLR. This is the map layer network node number by which the HLR identifies the Prime Access Registrar in the network. The MLC number is configured in E.164 format.</p> <p> <b>Note</b> MLC is a max-15 digit number.</p>
TrafficMode	Required; specify the traffic mode values for the HLR.
LoadShareMode	<p>Required; specify the load share mode for the HLR.</p> <p>When there is more than one associations with HLR, then the load sharing is set as Signaling Link Selection (SLS). SLS is done based on a simple round-robin basis.</p>
SCCPVariant	<p>The Signaling Connection Control Part (SCCP) variant of the Global Title:</p> <ul style="list-style-type: none"> <li>• Select <b>ITU88</b>, <b>ITU92</b>, or <b>ITU96</b>, if NetworkVariant is set to ITU.</li> <li>• Select <b>ANS88</b>, <b>ANS92</b>, or <b>ANS96</b>, if NetworkVariant is set to ANS.</li> </ul>
MaxTimeOuts	Maximum number of timeouts allowed for the remote server.
<b>RoutingParameters</b>	
OriginPointCode	Required; specify the originating point of a message in a signalling network. The value ranges from 0 - 16777215.
DestinationPointCode	Required; specify the destination address of a signalling point in a SS7 network.
RemoteSubSystemNumber	Required; specify the sub system number of the remote server. The RemoteSubSystemNumber is set as 6 by default.
OPCMask	Required; specify the wild card mask for the origin point code. The value ranges from 0 - 16777215.
DPCMask	Specify the wild card mask for the destination point code. The value ranges from 0 - 16777215.
ServiceIndicatorOctet	Specify the service identifier octet. The value ranges from 0 - 255.
RoutingContext	Required; specify the routing context which ranges from 0 - 16777215.
<b>Source &amp; Destination IP Addresses</b>	

**Table 2-69 Other Server Properties (continued)**

Fields	Description
SourceIPAddresses	Applicable only for Sigtran-m3ua protocol type. Enter the source IP address to be configured on the remote server and then click <b>Add</b> . The entered IP address is displayed in the SourceIPAddresses list box. Click <b>Delete</b> to remove the IP address from the list.
DestinationIPAddresses	Applicable only for Sigtran-m3ua protocol type. Enter the destination IP address to be configured on the remote server and then click <b>Add</b> . The entered IP address is displayed in the DestinationIPAddresses list box. Click <b>Delete</b> to remove the IP address from the list.

You can use the RemoteServers page allows for the following:

- [Filtering Records](#)
- [Setting Other Specifications](#)
- [Editing Records](#)
- [Deleting Records](#)

### Setting Other Specifications

To set up other specifications:

- 
- Step 1** Select **Network Resources > RemoteServers > Others**. The RemoteServers page is displayed.
  - Step 2** Click **Add** to add other specifications. The Remote Server Details page is displayed.
  - Step 3** Enter the required details.
  - Step 4** Click **Add Radius Server** to save the specified details in the Remote Server Details page. Otherwise click **Cancel** to return to the RemoteServers page without saving the details.

The RemoteServers page is displayed with the newly added details or a respective error message is displayed.

---

## Administration

Administration constitutes the maintenance and management of details specific administrator, various statistical data respective to the administrators, backing up and restoring server details, and license management of the server.

This section describes the following:

- [Administrators](#)
- [Statistics](#)
- [DiameterStatistics](#)
- [TACACStatistics](#)

- [Back Up and Restore](#)
- [LicenseUpload](#)

## Administrators

Prime Access Registrar provided *super-user* administrative access in which administrator can perform all tasks including starting and stopping the system and changing the configuration.

Prime Access Registrar also provides view-only administrative access. View-only access restricts an administrator to only being able to observe the system and prevents that user from making changes.

[Table 2-70](#) lists and describes the fields in the Administrator Details page.

**Table 2-70 Administrator Properties**

Fields	Description
Name	Required; administrator's user ID.
Description	Optional; description of the administrator.
New Password	Required; encrypted password of the administrator.
Confirm New Password	Required; encrypted password of the administrator and must match Password.
View Only	Default value (FALSE) indicates that the administrator is able to modify the configuration. When set to TRUE, the administrator can only view the server configuration and set the change the server trace level.

You can use the Administrators page for the following:

- [Filtering Records](#)
- [Adding Administrator Details](#)
- [Statistics](#)
- [Editing Records](#)
- [Deleting Records](#)

## Adding Administrator Details

To add new Administrator details:

- 
- Step 1** Choose **Administration > Administrators**. The Administrators page is displayed.
  - Step 2** Click **Add** to add administrator details. The Administrator Details page is displayed.
  - Step 3** Specify the required details.
  - Step 4** Click **Submit** to save the specified details in the Administrator Details page. Otherwise click **Cancel** to return to the Administrators page without saving the details.

The Administrators page is displayed with the newly added details or a respective error message is displayed.

---

## Statistics

This feature provides statistical information on the specified RADIUS server.

Table 2-71 lists the statistics information of the RADIUS server.

**Table 2-71** *aregcmd stats Information for RADIUS server*

Stats Value	Meaning
serverStartTime	Indicates the start time of the server.
serverResetTime	Indicates the time when the server was reloaded.
serverStat	Indicates if the server is running or stopped.
totalPacketsInPool	Number of packets that can be accommodated in the pool.
totalPacketsReceived	Number of packets that are received by RADIUS server.
totalPacketsSent	Number of packets that are sent by RADIUS server.
totalRequests	Number of requests received by RADIUS server. This includes access requests and accounting requests.
totalResponses	Number of responses sent by RADIUS server. This includes access accepts/rejects and accounting responses.
totalAccessRequests	Number of access requests received/processed by RADIUS server.
totalAccessAccepts	Number of access accepts sent by RADIUS server.
totalAccessChallenges	Number of access challenges sent by RADIUS server.
totalAccessRejects	Number of access rejects sent by RADIUS server.
totalAccessResponses	Number of access responses sent by RADIUS server.
totalAccountingRequests	Number of accounting requests received by RADIUS server.
totalAccountingResponses	Number of accounting responses sent by RADIUS server.
totalStatusServerRequests	Number of status server request received by RADIUS server.
totalAscendIPAAAllocateRequests	Number of requests received related to Ascend IP address allocation.

**Table 2-71** *aregcmd stats Information for RADIUS server (continued)*

<b>Stats Value</b>	<b>Meaning</b>
totalAscendIPAAlocateResponses	Number of responses sent related to Ascend IP Address Allocation.
totalAscendIPAReleaseRequests	Number of requests received related to Ascend IP Address release.
totalAscendIPAReleaseResponses	Number of responses sent related to Ascend IP Address release.
totalUSRNASRebootRequests	Number of user NAS reboot request received by RADIUS server.
totalUSRNASRebootResponses	Number of user NAS reboot response sent by RADIUS server.
totalUSRResourceFreeRequests	Number of user resource free request received by RADIUS server.
totalUSRResourceFreeResponses	Number of user resource free response sent by RADIUS server.
totalUSRQueryResourceRequests	Number of user query resource request received by RADIUS server.
totalUSRQueryResourceResponses	Number of user query resource response sent by RADIUS server.
totalUSRQueryReclaimRequests	Number of user query reclaim request received by RADIUS server.
totalUSRQueryReclaimResponses	Number of user query reclaim response sent by RADIUS server.
totalPacketsInUse	Number of packets that are being used.
totalPacketsDrained	Number of packets that are drained.
totalPacketsDropped	Number of packets that are dropped.
totalPayloadDecryptionFailures	Number of failures due to payloads decryption.
totalEAPSIMDecryptionFailures	Number of IMSI decryption failures for EAP-SIM services.
totalEAPSIMDecryptionSuccess	Number of IMSI decryption success for EAP-SIM services.
totalEAPAKADecryptionFailures	Number of IMSI decryption failures for EAP-AKA services.
totalEAPAKADecryptionSuccess	Number of IMSI decryption success for EAP-AKA services.
totalEAPAKAPRIMEDecryptionFailures	Number of IMSI decryption failures for EAP-AKA' services.
totalEAPAKAPRIMEDecryptionSuccess	Number of IMSI decryption success for EAP-AKA' services.
OCIActiveConnectionCount	Number of active OCI connections from Prime Access Registrar to the Oracle database.

**Table 2-71** *aregcmd stats Information for RADIUS server (continued)*

<b>Stats Value</b>	<b>Meaning</b>
TotalRESErrorResponses	Number of error responses from REST server.
TotalRequestsAcknowledged	Number of responses received since last server restart.
TotalResponsesDroppedForNotInCache	Number of responses dropped because their ID did not match the ID of any Pending requests.
TotalResponsesDroppedForSignatureMismatch	Number of responses dropped because their response authenticator did not decode to the correct shared secret.
TotalRequestsDroppedAfterMaxTries	Number of requests dropped because no response was received after retrying the configured number of times. This value is different from totalRequestsTimedOut because using the default configuration values, no response within 2000 ms bumps the TimedOut counter, but it waits 14000 ms (2000 + 4000 + 8000) to bump this counter.
LastRequestTime	Date and time of last proxy request.
LastAcceptTime	Date and time of last ACCEPT response to a client.
The following fields appear when you select a RADIUS client from the Clients drop-down list box at the bottom of the page.	
RADIUS Client statistics for:	Provides client's IP address, name, and IP address type
TLSActiveConnectionCount	Number of active TLS connections established for the RADIUS client.
totalAuthAccessRequests	Number of authentication access requests that are received by Prime Access Registrar from the client.
totalAuthDupAccessRequests	Number of duplicate authentication access requests that are received by Prime Access Registrar from the client.
totalAuthAccessAccepts	Number of authentication access requests from the client that are accepted by Prime Access Registrar.
totalAuthAccessRejects	Number of authentication access requests from the client that are rejected by Prime Access Registrar.
totalAuthAccessChallenges	Number of authentication challenges that are faced by Prime Access Registrar for the requests raised by the client.
totalAuthMalformedAccessRequests	Number of malformed authentication access requests that are received by Prime Access Registrar from the client.

**Table 2-71** *aregcmd stats Information for RADIUS server (continued)*

<b>Stats Value</b>	<b>Meaning</b>
totalAuthBadAuthenticators	Number of bad authentication access requests that are received by Prime Access Registrar from the client.
totalAuthPacketsDropped	Number of authentication access requests received from the client that are dropped by Prime Access Registrar. The packets, which are invalid and do not fulfill the parsing conditions, are dropped.
totalAuthUnknownTypes	Number of unknown authentication access requests that are received by Prime Access Registrar from the client.
totalAccPacketsDropped	Number of accounting access requests received from the client that are dropped by Prime Access Registrar. The packets, which are invalid and do not fulfill the parsing conditions, are dropped.
totalAccRequests	Number of accounting access requests received by Prime Access Registrar from the client.
totalAccDupRequests	Number of duplicate accounting access requests that are received by Prime Access Registrar from the client.
totalAccResponses	Number of accounting response sent by Prime Access Registrar to the client
totalAccBadAuthenticators	Number of bad accounting access requests that are received by Prime Access Registrar from the client.
totalAccMalformedRequests	Number of malformed accounting access requests that are received by Prime Access Registrar from the client.
totalAccNoRecords	Number of accounting access requests that are received with no records by Prime Access Registrar from the client.
totalAccUnknownTypes	Number of unknown accounting access requests that are received by Prime Access Registrar from the client.

## Resetting Server Statistics

To reset server statistics:

- 
- Step 1** Choose **Administration > Statistics**. The Radius Server Statistics page is displayed.
  - Step 2** Click **Reset** to reset all the RADIUS server statistics.
-

## DiameterStatistics

Prime Access Registrar supports statistic of Diameter messages through the CLI/GUI and SNMP. The existing 'stats' module has been extended to include additional counters related to Diameter. The Diameter statistics includes peer statistics and global summary statistics details on the specified server.

Table 2-72 lists the statistics information of the Diameter server. The statistical information in Table 2-73 is displayed based on the Diameter peer selected. Table 2-74 is displayed based on the Diameter remote server selected.

**Table 2-72** Diameter Stats Information

Metric	Value
<b>Diameter Statistics</b>	
serverStartTime	The start time of the server.
serverResetTime	The reset time of the server.
serverState	The state of the server.
cdbpLocalStatsTotalUpTime	The total time for which the Diameter server is up.
cdbpLocalResetTime	The time elapsed since a server was reset.
cdbpLocalStatsTotalPacketsIn	The total number of packets received by a Diameter Base protocol.
cdbpLocalStatsTotalPacketsOut	The total number of packets transmitted by a Diameter Base protocol.
cdbpLocalStatsTotalPacketsInUse	The total number of packets used.
Peer	The name of the peer. You can select a peer from the drop-down list.

**Table 2-73** Diameter Peer Stats Information

Metric	Value
<b>Diameter Peers:</b> To view the following fields, select a Diameter peer from the <b>Peer</b> drop-down list box and then click <b>Show Peer Stats</b> . Click <b>Reset</b> , to reset all the Diameter statistics of the peer.	
Stats for the Remote Server	The name of the selected peer.
ipaddress	The IP address of the peer.
port	The port of the peer.
cdbpPeerStatsState	Indicates the connection state in the Peer State Machine of the peer with which the Diameter server is communicating.
cdbpPeerStatsASAsOut	Number of Abort-Session-Answer messages that are sent to the peer.
cdbpPeerStatsACRsIn	Number of Accounting-Request messages that are received from the peer

**Table 2-73** Diameter Peer Stats Information (continued)

<b>Metric</b>	<b>Value</b>
cdbpPeerStatsACRsOut	Number of Accounting-Request messages that are sent to the peer.
cdbpPeerStatsACAsIn	Number of Accounting-Answer messages that are received from the peer.
cdbpPeerStatsACAsOut	Number of Accounting-Answer messages that are sent to the peer.
cdbpPeerStatsCERsIn	Number of Capabilities-Exchange-Request messages received from the peer.
cdbpPeerStatsCERsOut	Number of Capabilities-Exchange-Request messages sent to the peer.
cdbpPeerStatsCEAsIn	Number of Capabilities-Exchange-Answer messages received from the peer.
cdbpPeerStatsCEAsOut	Number of Capabilities-Exchange-Answer messages sent to the peer.
cdbpPeerStatsDWRsIn	Number of Device-Watchdog-Request messages received from the peer.
cdbpPeerStatsStateDuration	Represents the Peer state duration.
cdbpPeerStatsDWRsOut	Number of Device-Watchdog-Request messages sent to the peer.
cdbpPeerStatsDWAsIn	Number of Device-Watchdog-Answer messages received from the peer.
cdbpPeerStatsDWAsOut	Number of Device-Watchdog-Answer messages sent to the peer.
cdbpPeerStatsDPRsIn	Number of Disconnect-Peer-Request messages received from the peer.
cdbpPeerStatsDPRsOut	Number of Disconnect-Peer-Request messages sent to the peer.
cdbpPeerStatsDPAsIn	Number of Disconnect-Peer-Answer messages received from the peer.
cdbpPeerStatsDPAsOut	Number of Disconnect-Peer-Answer messages sent to the peer.
cdbpPeerStatsRARsIn	Number of Re-Auth-Request messages that are received from the peer.
cdbpPeerStatsRARsOut	Number of Re-Auth-Request messages that are sent to the peer.
cdbpPeerStatsRAAsIn	Number of Re-Auth-Answer messages that are received from the peer.
cdbpPeerStatsLastDiscCause	The last cause for a peer's disconnection.
cdbpPeerStatsRAAsOut	Number of Re-Auth-Answer messages that are sent to the peer.
cdbpPeerStatsSTRsIn	Number of Session-Termination-Request messages that are received from the peer.

**Table 2-73** *Diameter Peer Stats Information (continued)*

<b>Metric</b>	<b>Value</b>
cdbpPeerStatsSTRsOut	Number of Session-Termination-Request messages that are sent to the peer.
cdbpPeerStatsSTAsIn	Number of Session-Termination-Answer messages that are received from the peer.
cdbpPeerStatsSTAsOut	Number of Session-Termination-Answer messages that are sent to the peer.
cdbpPeerStatsDWReqTimer	The interval between the packets that are sent to the peers.
cdbpPeerStatsRedirectEvents	Number of redirects that are sent from a peer.
cdbpPeerStatsAccDupRequests	Number of duplicate Diameter Accounting-Request packets.
cdbpPeerStatsMalformedReqsts	Number of malformed Diameter packets that are received.
cdbpPeerStatsAccsNotRecorded	Number of Diameter Accounting-Request packets that are received and responded but not recorded.
cdbpPeerStatsWhoInitDisconnect	Indicates whether the host or peer initiated the disconnect.
cdbpPeerStatsAccRetrans	Number of Diameter Accounting-Request packets that are retransmitted to the Diameter server.
cdbpPeerStatsTotalRetrans	Number of Diameter packets that are retransmitted to the Diameter server. This does not include the Diameter Accounting-Request packets that are retransmitted.
cdbpPeerStatsAccPendReqstsOut	Number of Diameter Accounting-Request packets that are sent to the peer which have not yet timed out or received a response. This variable is incremented when an Accounting-Request is sent to the server and decremented due to receipt of an Accounting-Response, a timeout or a retransmission.
cdbpPeerStatsAccReqstsDropped	Number of Accounting-Requests to the server that are dropped.
cdbpPeerStatsHByHDropMessages	An answer message that is received with an unknown hop-by-hop identifier. This does not include the accounting requests that are dropped.
cdbpPeerStatsEToEDupMessages	The duplicate answer messages that are locally consumed. This does not include duplicate accounting requests that are received.

**Table 2-73** Diameter Peer Stats Information (continued)

<b>Metric</b>	<b>Value</b>
<code>cdpPeerStatsUnknownTypes</code>	Number of Diameter packets of unknown type that are received from the peer.
<code>cdpPeerStatsProtocolErrors</code>	Number of protocol errors that are returned to peer, but not including the redirects.
<code>cdpPeerStatsTransientFailures</code>	Indicates the transient failure count.
<code>cdpPeerStatsPermanentFailures</code>	Indicates the permanent failure count.
<code>cdpPeerStatsDWCurrentStatus</code>	Indicates the connection status of the peer.
<code>cdpPeerStatsTransportDown</code>	Number of unexpected transport failures.
<code>cdpPeerStatsTimeoutConnAtmpts</code>	Number of times the server attempts to connect to a peer when there is no transport connection with the peer. This is reset on disconnection.
<code>cdpPeerStatsASRsIn</code>	Number of Abort-Session-Request messages that are received from the peer.
<code>cdpPeerStatsASRsOut</code>	Number Abort-Session-Request messages that are sent to the peer.
<code>cdpPeerStatsASAsIn</code>	Number of Abort-Session-Answer messages that are received from the peer.
<code>cdpPeerStatsDERsIn</code>	Number of Diameter-EAP-Request messages that are received from the peer.
<code>cdpPeerStatsDERsOut</code>	Number of Diameter-EAP-Request messages that are sent to the peer.
<code>cdpPeerStatsDEAsIn</code>	Number of Diameter-EAP-Answer messages that are received from the peer.
<code>cdpPeerStatsDEAsOut</code>	Number of Diameter-EAP-Answer messages that are sent to the peer.
<code>cdpPeerStatsAARsIn</code>	Number of AA-Request messages that are received from the peer.
<code>cdpPeerStatsAARsOut</code>	Number of AA-Request messages that are sent to the peer.
<code>cdpPeerStatsAAAsIn</code>	Number of AA-Answer messages that are received from the peer.
<code>cdpPeerStatsAAAsOut</code>	Number of AA-Answer messages that are sent to the peer.
<code>cdpPeerStatsMARsIn</code>	Number of Multimedia-Authentication-Request messages that are received from the peer.
<code>cdpPeerStatsMARsOut</code>	Number of Multimedia-Authentication-Request messages that are sent to the peer.
<code>cdpPeerStatsMAAsIn</code>	Number of Mutlimedia-Authentication-Answer messages that are received from the peer.

**Table 2-73** *Diameter Peer Stats Information (continued)*

<b>Metric</b>	<b>Value</b>
cdbpPeerStatsMAAsOut	Number of Multimedia-Authentication-Answer messages that are sent to the peer.
cdbpPeerStatsSARsIn	Number of Server-Assignment-Request messages that are received from the peer.
ccdbpPeerStatsSARsOut	Number of Server-Assignment-Request messages that are sent to the peer.
cdbpPeerStatsSAAsIn	Number of Server-Assignment-Answer messages that are received from the peer.
cdbpPeerStatsSAAsOut	Number of Server-Assignment-Answer messages that are sent to the peer.
cdbpPeerStatsRTRsIn	Number of Registration-Termination-Request messages that are received from the peer.
cdbpPeerStatsRTRsOut	Number of Registration-Termination-Request messages that are sent to the peer.
cdbpPeerStatsRTAsIn	Number of Registration-Termination-Answer messages that are received from the peer.
cdbpPeerStatsRTAsOut	Number of Registration-Termination-Answer messages that are sent to the peer.
cdbpPeerStatsPPRsIn	Number of Push-Profile-Request messages that are received from the peer.
cdbpPeerStatsPPRsOut	Number of Push-Profile-Request messages that are sent to the peer.
cdbpPeerStatsPPAsIn	Number of Push-Profile-Answer messages that are received from the peer.
cdbpPeerStatsPPAsOut	Number of Push-Profile-Answer messages that are sent to the peer.

**Table 2-74** *Diameter Remote Server Stats Information*

<b>Metric</b>	<b>Value</b>
<b>Diameter RemoteServers:</b> To view the following fields, select a remote server from the <b>Remote-Servers</b> drop-down list box and then click <b>Show RemoteServer Stats</b> . Click <b>Reset</b> , to reset all the Diameter statistics of the remote server.	
Stats for the Remote Server	The name of the selected remote server.
ipaddress	The IP address of the remote server.
port	The port of the remote server.
cDiaRemSvrActive	Indicates whether the server was active (not in a down state).
cDiaRemSvrRTTAverage	Average round trip time since the last server restart.

**Table 2-74** *Diameter Remote Server Stats Information (continued)*

<b>Metric</b>	<b>Value</b>
cDiaRemSvrRTTDeviation	Indicates a standard deviation of the RTTAverage.
cDiaRemSvrServerType	Indicates the remote server type.
cDiaRemSvrTotalRequestsPending	Number of requests currently queued.
cDiaRemSvrTotalRequestsOutstanding	Number of requests currently proxied that have not yet returned
cDiaRemSvrTotalRequestsAcknowledged	Number of responses received since last server restart.
cDiaRemSvrStatsState	Indicates the connection state of the Diameter remote server.
cDiaRemSvrStatsASRsIn	Number of Abort-Session-Request messages that are received by the remote server.
cDiaRemSvrStatsASRsOut	Number Abort-Session-Request messages that are sent by the remote server.
cDiaRemSvrStatsASAsIn	Number of Abort-Session-Answer messages that are received by the remote server.
cDiaRemSvrStatsASAsOut	Number of Abort-Session-Answer messages that are sent by the remote server.
cDiaRemSvrStatsACRsIn	Number of Accounting-Request messages that are received by the remote server.
cDiaRemSvrStatsACRsOut	Number of Accounting-Request messages that are sent by the remote server.
cDiaRemSvrStatsACAsIn	Number of Accounting-Answer messages that are received by the remote server.
cDiaRemSvrStatsACAsOut	Number of Accounting-Answer messages that are sent by the remote server.
cDiaRemSvrStatsCERsIn	Number of Capabilities-Exchange-Request messages received by the remote server.
cDiaRemSvrStatsCERsOut	Number of Capabilities-Exchange-Request messages sent by the remote server.
cDiaRemSvrStatsCEAsIn	Number of Capabilities-Exchange-Answer messages received by the remote server.
cDiaRemSvrStatsCEAsOut	Number of Capabilities-Exchange-Answer messages sent by the remote server.
cDiaRemSvrStatsDWRsIn	Number of Device-Watchdog-Request messages received by the remote server.
cDiaRemSvrStatsDWRsOut	Number of Device-Watchdog-Request messages sent by the remote server.
cDiaRemSvrStatsDWAsIn	Number of Device-Watchdog-Answer messages received by the remote server.
cDiaRemSvrStatsDWAsOut	Number of Device-Watchdog-Answer messages sent by the remote server.

**Table 2-74** *Diameter Remote Server Stats Information (continued)*

<b>Metric</b>	<b>Value</b>
cDiaRemSvrStatsDPRsIn	Number of Disconnect-Peer-Request messages received by the remote server.
cDiaRemSvrStatsDPRsOut	Number of Disconnect-Peer-Request messages sent by the remote server.
cDiaRemSvrStatsDPAsIn	Number of Disconnect-Peer-Answer messages received by the remote server.
cDiaRemSvrStatsDPAsOut	Number of Disconnect-Peer-Answer messages sent by the remote server.
cDiaRemSvrStatsRARsIn	Number of Re-Auth-Request messages that are received by the remote server.
cDiaRemSvrStatsRARsOut	Number of Re-Auth-Request messages that are sent by the remote server.
cDiaRemSvrStatsRAAsIn	Number of Re-Auth-Answer messages that are received by the remote server.
cDiaRemSvrStatsRAAsOut	Number of Re-Auth-Answer messages that are sent by the remote server.
cDiaRemSvrStatsSTRsIn	Number of Session-Termination-Request messages that are received by the remote server.
cDiaRemSvrStatsSTRsOut	Number of Session-Termination-Request messages that are sent by the remote server.
cDiaRemSvrStatsSTAsIn	Number of Session-Termination-Answer messages that are received by the remote server.
cDiaRemSvrStatsSTAsOut	Number of Session-Termination-Answer messages that are sent by the remote server.
cDiaRemSvrStatsRedirectEvents	Number of redirects that are sent from the remote server.
cDiaRemSvrStatsAccDupRequests	Number of duplicate Diameter Accounting-Request packets.
cDiaRemSvrStatsMalformedRequests	Number of malformed Diameter packets that are received.
cDiaRemSvrStatsAccsNotRecorded	Number of Diameter Accounting-Request packets that are received and responded but not recorded.
cDiaRemSvrStatsWhoInitDisconnect	Indicates whether the host or remote server initiated the disconnect.
cDiaRemSvrStatsAccRetrans	Number of Diameter Accounting-Request packets that are retransmitted by the Diameter remote server.

**Table 2-74** *Diameter Remote Server Stats Information (continued)*

<b>Metric</b>	<b>Value</b>
cDiaRemSvrStatsTotalRetrans	Number of Diameter packets that are retransmitted by the Diameter server. This does not include the Diameter Accounting-Request packets that are retransmitted.
cDiaRemSvrStatsAccPendRequestsOut	Number of Diameter Accounting-Request packets that are sent by the remote server which have not yet timed out or received a response. This variable is incremented when an Accounting-Request is sent by the server and decremented due to receipt of an Accounting-Response, a timeout or a retransmission.
cDiaRemSvrStatsAccReqstsDropped	Number of Accounting-Requests that are dropped.
cDiaRemSvrStatsHByHDropMessages	An answer message that is received with an unknown hop-by-hop identifier. This does not include the accounting requests that are dropped.
cDiaRemSvrStatsEToEDupMessages	The duplicate answer messages that are locally consumed. This does not include duplicate accounting requests that are received.
cDiaRemSvrStatsUnknownTypes	Number of Diameter packets of unknown type that are received by the remote server.
cDiaRemSvrStatsProtocolErrors	Number of protocol errors that are returned by the remote server, but not including the redirects.
cDiaRemSvrStatsTransientFailures	Indicates the transient failure count.
cDiaRemSvrStatsPermanentFailures	Indicates the permanent failure count.
cDiaRemSvrStatsDWCurrenStatus	Indicates the connection status of the remote server.
cDiaRemSvrStatsTransportDown	Number of unexpected transport failures.
cDiaRemSvrStatsTimeoutConnAtmpts	Number of times the remote server attempts to reconnect when there is no transport connection. This is reset on disconnection.
cDiaRemSvrStatsMARsIn	Number of Multimedia-Authentication-Request messages that are received by the remote server.
cDiaRemSvrStatsMARsOut	Number of Multimedia-Authentication-Request messages that are sent by the remote server.
cDiaRemSvrStatsMAAsIn	Number of Mutlimedia-Authentication-Answer messages that are received by the remote server.

**Table 2-74** *Diameter Remote Server Stats Information (continued)*

<b>Metric</b>	<b>Value</b>
cDiaRemSvrStatsMAAsOut	Number of Multimedia-Authentication-Answer messages that are sent by the remote server.
cDiaRemSvrStatsSARsIn	Number of Server-Assignment-Request messages that are received by the remote server.
cDiaRemSvrStatsSARsOut	Number of Server-Assignment-Request messages that are sent by the remote server.
cDiaRemSvrStatsSAAsIn	Number of Server-Assignment-Answer messages that are received by the remote server.
cDiaRemSvrStatsSAAsOut	Number of Server-Assignment-Answer messages that are sent by the remote server.
cDiaRemSvrStatsRTRsIn	Number of Registration-Termination-Request messages that are received by the remote server.
cDiaRemSvrStatsRTRsOut	Number of Registration-Termination-Request messages that are sent by the remote server.
cDiaRemSvrStatsRTAsIn	Number of Registration-Termination-Answer messages that are received by the remote server.
cDiaRemSvrStatsRTAsOut	Number of Registration-Termination-Answer messages that are sent by the remote server.
cDiaRemSvrStatsPPRsIn	Number of Push-Profile-Request messages that are received by the remote server.
cDiaRemSvrStatsPPRsOut	Number of Push-Profile-Request messages that are sent by the remote server.
cDiaRemSvrStatsPPAsIn	Number of Push-Profile-Answer messages that are received by the remote server.
cDiaRemSvrStatsPPAsOut	Number of Push-Profile-Answer messages that are sent by the remote server.
cDiaRemSvrStatsDERsIn	Number of Diameter-EAP-Request messages that are received by the remote server.
cDiaRemSvrStatsDERsOut	Number of Diameter-EAP-Request messages that are sent by the remote server.
cDiaRemSvrStatsDEAsIn	Number of Diameter-EAP-Answer messages that are received by the remote server.
cDiaRemSvrStatsDEAsOut	Number of Diameter-EAP-Answer messages that are sent by the remote server.
cDiaRemSvrStatsAARsIn	Number of AA-Request messages that are received by the remote server.
cDiaRemSvrStatsAARsOut	Number of AA-Request messages that are sent by the remote server.

**Table 2-74** *Diameter Remote Server Stats Information (continued)*

<b>Metric</b>	<b>Value</b>
cDiaRemSvrStatsAAAsIn	Number of AA-Answer messages that are received by the remote server.
cDiaRemSvrStatsAAAsOut	Number of AA-Answer messages that are sent by the remote server.

## TACACSStatistics

Prime Access Registrar supports CISCO-AAA-SERVER-MIB to describe the statistics of TACACS+ protocol. This is supported through CLI/GUI and SNMP.

Table 2-75 lists the statistics information and the meaning of the values.

**Table 2-75** *TACACS Stats Information*

<b>Metric</b>	<b>Value</b>
<b>TACACS Statistics</b>	
serverStartTime	The start time of the server.
serverResetTime	The reset time of the server.
serverState	The state of the server.
totalPacketsReceived	Number of packets that are received by a TACACS+ protocol irrespective of the type of Authentication and Accounting.
totalPacketsSent	Number of packets that are sent by a TACACS+ protocol irrespective of the type of Authentication and Accounting.
totalRequests	Number of packet requests that are received by a TACACS+ protocol irrespective of the type of Authentication and Accounting.
totalResponses	Number of packet responses that are sent by a TACACS+ protocol irrespective of the type of Authentication and Accounting.
totalAuthenticationRequests	Number of authentication requests that are received by Prime Access Registrar.
totalAuthenticationAccepts	Number of authentication requests that are accepted by Prime Access Registrar.
totalAuthenticationRejects	Number of authentication requests that are rejected by Prime Access Registrar.
totalAuthenticationChallenges	Number of authentication challenges that are faced by Prime Access Registrar.
totalAuthenticationResponses	Number of authentication responses that are sent by Prime Access Registrar.
totalAuthorizationRequests	Number of authorization requests that are received by Prime Access Registrar.

**Table 2-75 TACACS Stats Information (continued)**

<b>Metric</b>	<b>Value</b>
totalAuthorizationAccepts	Number of authorization requests that are accepted by Prime Access Registrar.
totalAuthorizationRejects	Number of authorization requests that are rejected by Prime Access Registrar.
totalAuthorizationResponses	Number of authorization responses that are sent by Prime Access Registrar.
totalAccountingRequests	Number of accounting requests that are received by Prime Access Registrar.
totalAccountingAccepts	Number of accounting requests that are accepted by Prime Access Registrar.
totalAccountingRejects	Number of accounting requests that are rejected by Prime Access Registrar.
totalAccountingResponses	Number of accounting requests that are sent by Prime Access Registrar.
totalPayloadDecryptionFailures	Number of packets that are not decrypted by Prime Access Registrar.
totalPacketsDropped	Number of packets that are dropped by Prime Access Registrar. The packets are dropped, which are invalid and do not fulfill the parsing conditions.

## Back Up and Restore

To back up and restore the server details, Choose **Administration > Backup&Restore**. The Backup page is displayed with the list of recently backed up details of the server with the date and time. This option allows you to take a backup of the database, sessions, and scripts, and stores it in **/cisco-ar/backup** directory.

### Backup Server Details

To back up the server details:

- 
- Step 1** Choose **Administration > Backup & Restore**. The Backup page is displayed.
  - Step 2** Click **Backup** to take a backup of the database, sessions, and scripts, and stores it in **/cisco-ar/backup** directory. The details will be backed up and appended to the backup list and displayed in the Backup page.
- 

### Restoring Server Details

To restore the backed-up server details:

- 
- Step 1** Choose **Administration > Backup & Restore**. The Backup page is displayed.
  - Step 2** Choose the record from the backup list.

- Step 3** Click **Restore**. The details of the selected back up file will be restored successfully.
- 

## LicenseUpload

Prime Access Registrar license information are uploaded using the Upload feature. To upload the license file:

### Uploading License File

To upload the Prime Access Registrar license file:

- 
- Step 1** Choose **Administration > LicenseUpload**. The Prime Access Registrar License-Upload page is displayed.
- Step 2** Click **Browse** to locate the license file. The File Upload dialog box is displayed.
- Step 3** Choose the required file.
- Step 4** Click **Upload**. The selected file will be uploaded in **/cisco-ar/license** directory.



---

**Note** You need to ensure that the license file that you want to upload should be in **.lic** format.

---

- Step 5** Click **Reset** to clear the text in the Select the File field, if you want to clear the selected path.
- 

## Read-Only GUI

Prime Access Registrar provides a read-only GUI that enables an administrator to observe the system but prevents that administrator from making changes.

When you configure a user to be an administrator, check the View-Only check box to limit the administrator to view-only operation. You can also use the CLI by setting the View-Only property to TRUE under /Administrator/admin\_name.

When using the Read-Only GUI, the Configuration, Network Resources and Administration sections are displayed as same as a fully-enabled administrator. The details of these sections are displayed in text format and cannot be edited.





## RADIUS Accounting

---

This chapter describes RADIUS Accounting in Cisco Prime Access Registrar (Prime Access Registrar) as defined in Internet RFC 2866.

This chapter contains the following sections:

- [Understanding RADIUS Accounting](#)
- [Setting Up Accounting](#)
- [Oracle Accounting](#)
- [LDAP Accounting](#)
- [MySQL Support](#)
- [Proxying Accounting Records](#)

## Understanding RADIUS Accounting

RADIUS accounting is the process of collecting and storing the information contained in

- Accounting-Start and
- Accounting-Stop messages.

Internet RFC 2866 describes the protocol for sending accounting information between a Network Access Server (NAS) and a RADIUS server (or shared accounting server).



**Note**

---

Prime Access Registrar uses UDP port number 1813 as its default port for RADIUS accounting messages. RFC 2866 defines UDP port number 1813 as the accounting port number.

---

When a NAS that uses accounting begins a session, it sends an Accounting-Start packet describing the type of service and the user being connected to the Prime Access Registrar server. When the session ends, the NAS sends the RADIUS server an Accounting Stop packet describing the type of service that was delivered. The Accounting Stop packet might also contain statistics such as elapsed time, input and output octets, or input and output packets.

# Setting Up Accounting

To configure Prime Access Registrar to perform accounting, you must do the following:

1. Create a service
2. Set the service type to file
3. Set the `DefaultAccountingService` field in `/Radius` to the name of the service you created

After you **save** and **reload** the Prime Access Registrar server configuration, the Prime Access Registrar server writes accounting messages to the **accounting.log** file in the `/opt/CSCOar/logs` directory. The Prime Access Registrar server stores information in the **accounting.log** file until a rollover event occurs. A rollover event is caused by the **accounting.log** file exceeding a pre-set size, a period of time transpiring, or on a scheduled date.



## Note

You can also choose to export the accounting messages to a .csv file by providing the appropriate file type in the accounting service.

When the rollover event occurs, the data in **accounting.log** is stored in a file named by the prefix *accounting*, a date stamp (*yyyymmdd*), and the number of rollovers for that day. For example, **accounting-20131107-14** would be the 14th rollover on November 07, 2013.

The following shows the properties for a service called `CiscoAccounting`:

```
[ //localhost/Radius/Services/acc ]
  Name = acc
  Description =
  Type = file
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
  OutageScript~ =
  FilenamePrefix = accounting
  FileType~ = log
  EnableRollOverIntelligence = TRUE
  MaxFileSize = "10 Megabytes"
  MaxFileAge = "1 Day"
  RolloverSchedule =
  UseLocalTimeZone = FALSE
  AttributesToBeLogged/
    1. Acct-Session-Id
```

## Accounting Log File Rollover

The Prime Access Registrar accounting functionality provides flexibility in managing the accounting log. You can configure the Prime Access Registrar server to rollover the accounting log using any combination of the following Prime Access Registrar accounting service properties:

- `MaxFileSize`—Indicates the maximum size of the accounting log file in KB, MB, or GB
- `MaxFileAge`—Indicates the maximum age of the log file in minutes, hours, days, or weeks
- `RolloverSchedule`—Indicates the exact time including the day of the month or day of the week, hour and minute to roll over the accounting log file

You can configure an accounting service using any combination of `MaxFileSize`, `MaxFileAge`, and `RolloverSchedule`. For example, you might configure `RolloverSchedule` and `MaxFileAge` at the same time. This would be useful if you wanted to have an age-based rollover, but also synchronize to an absolute clock at specified times. The following would set a rollover every twelve hours at 11:59 and 12:59.

```
set MaxFileAge "12 H"
```

```
set RolloverSchedule "59 11,12 * * *"
```

You might also consider scheduling `MaxFileAge` to be six minutes and set `RolloverSchedule` to the top of the hour. The following would create ten six-minute long files starting anew every hour.

```
set MaxFileAge "6 Minutes"
```

```
set RolloverSchedule "0 * * * *"
```

Although you specify an exact time with the `RolloverSchedule` property, the Prime Access Registrar server only checks the rollover schedule when an accounting event occurs. If your Prime Access Registrar server receives a steady flow of packets (at least one per minute), the times you specify are accurate. However, if the Prime Access Registrar server does not receive any packets for a period of time, no rollovers will occur until the next packet is received. The same is true for `MaxFileAge` and `MaxFileSize`.

Based on the maximum file size and the age specified, Prime Access Registrar closes the accounting file, moves it to a new name, and reopens the file as a new file. The name given to this accounting file depends on its creation and modification dates.

For example, if the file was created and modified on the same date, the filename will be of the format `FileNamePrefix-<yyyymmdd>-<n>.log`, and the suffix will have year, month, day, and number. If the file was created on some day and modified on another, the filename will be of the format `FileNamePrefix-<yyyymmdd>-<yyyymmdd>-<n>.log`, and the suffix will have creation date, modification date, and number.

This section contains the following topics:

- [FilenamePrefix](#)
- [MaxFileSize](#)
- [MaxFileAge](#)
- [RolloverSchedule](#)
- [UseLocalTimeZone](#)
- [FileType](#)
- [EnableRolloverIntelligence](#)
- [AttributesToBeLogged](#)

## FilenamePrefix

The `FileNamePrefix` property enables you to specify a path to the file system in which you store the log files. If you do not manage your log files regularly, they might use the system resources, which will affect the performance of the Prime Access Registrar server.

We recommend that you store the log files in a file system different from the file system where you installed the Prime Access Registrar software by specifying the path in the `FilenamePrefix` property. By doing so the Prime Access Registrar server continues to run, even if the accounting logs fill the file system.

The following example specifies the `/usr/arlogs/accounting` as the `FilenamePrefix`:

```
set /Radius/Services/CiscoAccounting/FilenamePrefix /usr/arlogs/accounting
```

You can also set up a *cron job* to check the size of the log files and mail the administrator if the file system is full.

## MaxFileSize

Use `MaxFileSize` to indicate the maximum size of the **accounting.log** file in minutes, hours, days, or weeks. `MaxFileAge` measures the age of the **accounting.log** file from the time the previous file rollover occurred.

You can specify the following (case insensitive) file sizes:

- K, Kilobytes, Kilobytes
- M, Megabyte, Megabytes
- G, Gigabyte, Gigabytes

The following are examples of valid commands to set `MaxFileSize`:

```
set MaxFileSize "500 kilobytes"
```

The example above sets a `MaxFileSize` of 500 kilobytes

```
set maxfilesize "1 G"
```

The example above sets a `MaxFileSize` of one gigabyte

```
set maxfilesize "200 megabyte"
```

The example above sets a `MaxFileSize` of 200 megabytes

## MaxFileAge

Use `MaxFileAge` to indicate the maximum age of the log file in minutes, hours, days, or weeks. `MaxFileAge` measures the age of the **accounting.log** file from the time the previous file rollover occurred.

You can specify the following (case insensitive) periods of time:

- M, Minute, or Minutes preceded by a number from 0 to 59
- H, Hour, or Hours preceded by a number from 0 to 12
- D, Day, or Days preceded by a number from 1 to 31
- W, Week, or Weeks preceded by a number from 1 to 52

The following are examples of valid commands to set `MaxFileAge`:

```
set MaxFileAge "6 Minutes"
```

The example above sets a `MaxFileAge` of 6 minutes.

```
set maxfileage "2 d"
```

The example above sets a MaxFileAge of two days.

```
set maxfileage "1 H"
```

The example above sets a MaxFileAge of one hour.

## RolloverSchedule

You set RolloverSchedule using the following crontab-style time format:

```
minute hour "day of month" "month of year" "day of week"
```

Where:

- Minute is a value from 0-59
- Hour is a value from 0-12
- Day (of the month) is a value from 1-31
- Month is a value from 1-12
- Day (of the week) is a value from 0-6, where 0 is Sunday

## UseLocalTimeZone

When set to **TRUE**, the Prime Access Registrar server stores the accounting records in the log using the local system time. When set to **FALSE** (the default), Prime Access Registrar stores the accounting records in the log using Greenwich Mean Time (GMT).

## FileType

Use **FileType** to indicate the type of the file to export the accounting messages to. FileType could be one of the following:

- **log**— Prime Access Registrar server writes accounting messages to the accounting.log file in the /opt/CSCOAr/logs directory.
- **csv**—Prime Access Registrar server writes accounting messages to the accounting.csv file in the /opt/CSCOAr/logs directory. You must set up a delimiter for this file type, which could be ‘;’, ‘,’, and ‘:’.

## EnableRolloverIntelligence

When set to **TRUE**, rollover intelligence will be enabled for the accounting records based on the accounting service properties. For example, if a log file is deleted, this parameter will indicate whether to create a log with the deleted index before continuing with new indexes or to ignore the deleted index and create log files from the last index available for that date.

For example, if:

- there are log files such as **acct-1-1209-2015.log**, **acct-2-1209-2015.log**, through **acct-10-1209-2015.log** for that date
- **EnableRolloverIntelligence** is set to **TRUE**
- **acct-2-1209-2015.log** is deleted

The service creates a log file **acct-2-1209-2015.log** before continuing with **acct-11-1209-2015.log**.

If **EnableRolloverIntelligence** is set to **FALSE**, the service ignores **acct-2-1209-2015.log** and continues creating log files from **acct-11-1209-2015.log**.

## AttributesToBeLogged

The **AttributesToBeLogged** parameter allows you to configure the set of attributes that must be logged by the accounting file service for a particular packet. If this list is empty, the accounting file service logs all the attributes available for that particular packet.

# Oracle Accounting

Previous releases of Prime Access Registrar supported accessing user data from an Oracle database using Open Database Connectivity (ODBC), but this feature was limited to performing authentication and authorization (AA). You could only write the accounting records to local file or proxy to another RADIUS server.

Prime Access Registrar supports writing accounting records into Oracle database enabling integration between billing systems and Oracle.

- Prime Access Registrar adds a new type of service and remote server called *odbc-accounting* that enables inserting accounting records into Oracle.
- You can write accounting records into Oracle by referring this service in **/Radius/DefaultAccountingService** or in the Accounting-Service environment variable.

There is no specified schema structure to use the Oracle accounting feature. You can use your own table design and configure insert statements using standard SQL in the Prime Access Registrar configuration. The Prime Access Registrar server executes the insert statements to write the accounting record into Oracle. This feature is similar to the existing ODBC feature which performs authentication and authorization.

To improve latency for writing accounting records into database, packet buffering can be used. This option is enabled using the *BufferAccountingPackets* property under the *odbc-accounting* remote server definition.



---

**Note**

Prime Access Registrar supports Oracle 10g client and 11g server.

---



---

**Note**

For more information about dynamic SQL feature, see [Dynamic SQL Feature, page 3-11](#).

---

This section contains the following topics:

- [Configuring Oracle Accounting](#)
- [Packet Buffering](#)
- [Dynamic SQL Feature](#)

## Configuring Oracle Accounting

To use the Oracle accounting feature,

- you must configure a service of type *odbc-accounting* under **/Radius/Services**.
- you must also configure at least one remote servers of type *odbc-accounting* under **/Radius/RemoteServers**.

This section contains the following topics:

- [ODBC-Accounting Service](#)
- [Configuring Oracle Accounting](#)
- [ODBC RemoteServers](#)
- [Configuration Examples](#)
- [Packet Buffering](#)
- [Dynamic SQL Feature](#)

### ODBC-Accounting Service

The following is an example of an ODBC-Accounting service:

```
[ //localhost/Radius/Services/oracle_accounting ]
  Name = oracle_accounting
  Description =
  Type = odbc-accounting
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
  OutageScript~ =
  MultipleServersPolicy = Failover
  RemoteServers/
    1. accounting_server
```

### ODBC RemoteServers

Create a remote server under **/Radius/RemoteServers**, and set its protocol to *odbc-accounting*. The following is an example of an ODBC-Accounting RemoteServer's configuration:

```
[ //localhost/Radius/RemoteServers/accounting_server ]
  Name = accounting_server
  Description =
  Protocol = odbc-accounting
  ReactivateTimerInterval = 300000
  Timeout = 15
  DataSourceConnections = 8
  ODBCDataSource =
  KeepAliveTimerInterval = 0
  BufferAccountingPackets = TRUE
  MaximumBufferFileSize = "10 Megabytes"
  NumberOfRetriesForBufferedPacket = 3
  BackingStoreEnvironmentVariables =
  UseLocalTimeZone = FALSE
  AttributeList =
  Delimiter =
  SQLDefinition/
```

Table 3-1 describes the ODBC RemoteServer properties.

**Table 3-1 ODBC RemoteServer Properties**

Property	Description
Name	Name of the remote server; this property is mandatory, and there is no default
Description	Optional description of server
Protocol	Must be set to <code>odbc-accounting</code>
ReactivateTimerInterval	Mandatory time interval (in milliseconds) to activate an inactive server; defaults to 300000 ms.
Timeout	Mandatory time interval (in seconds) to wait for SQL operation to complete; defaults to 15 seconds
DataSourceConnections	Mandatory number of connections to be established; defaults to 8
ODBCDataSource	Name of the ODBCDataSource to use and must refer to one entry in the list of ODBC datasources configured under <b>/Radius/Advanced/ODBCDataSources</b> . Mandatory; no default
KeepAliveTimerInterval	Mandatory time interval (in milliseconds) to send a keepalive to keep the idle connection active; defaults to zero (0) meaning the option is disabled
BufferAccountingPackets	Mandatory, TRUE or FALSE, determines whether to buffer the accounting packets to local file, defaults to TRUE which means that packet buffering is enabled
MaximumBufferFileSize	Mandatory if BufferAccountingPackets is set to TRUE, determines the maximum buffer file size, defaults to 10 Megabyte)
NumberOfRetriesForBufferedPacket	Mandatory if BufferAccountingPackets is set to TRUE. A number greater than zero determines the number of attempts to be made to insert the buffered packet into Oracle. Defaults to 3.
BackingStoreEnvironmentVariables	Optional; when BufferAccountingPackets is set to TRUE, contains a comma-separated list of environment variable names to be stored into a local file along with buffered packet. No default. BackingStoreEnvironmentVariables can also be specified in scripts using the BackingStoreEnvironmentVariables environment variable.
UseLocalTimeZone	Set to TRUE or FALSE, determines the timezone of accounting records' TimeStamp (defaults to FALSE).
AttributeList	List of comma-separated attribute names.
Delimiter	Character used to separate the values of the attributes given in AttributeList property.
SQLDefinition	List of insert, update and delete statements to be executed to insert, update and delete the accounting record.

It is mandatory to set MaximumBufferFileSize property if BufferAccountingPackets property is set to TRUE. MaximumBufferFileSize can be specified in Kilobytes, Megabytes and Gigabytes. All values "512 kilobytes", "512 k", "512 KB" are valid for specifying 512 kilobytes.

If buffering is enabled, incoming packets will be accepted and logged to local file until the configured buffer file size is reached even if the database is offline. Attempts to insert them into Oracle will be made when database becomes available. This remote server will be marked as down only when the buffer gets

full. So, having two odbc-accounting remote servers in the service, first one with buffering enabled and multiple server policy of FailOver will make the other remote servers to receive packets only when the first remote server's buffer gets full.

AttributeList is to specify the list of attribute names separated with comma. When this 'AttributeList' is given in the MarkerList, these attributes' values will be appended together with delimiter specified in 'Delimiter' property and will be supplied as input to that marker.

Attributes from the Prime Access Registrar environment and request dictionaries can be specified in the MarkerList. Request dictionary will be looked up first for the attributes. Other than the standard attributes in the Prime Access Registrar dictionaries, two new marker variables are supported inside the marker list. They are,

- **TimeStamp**—Used to insert the timestamp into Oracle from Prime Access Registrar. Specifying this will supply the timestamp of that accounting record as a value to the insert statement. Time zone of this timestamp will be local if UseLocalTimeZone property is set to TRUE, otherwise GMT. This functionality could also be achieved by employing a trigger on the accounting table in the database. However, using this marker variable is recommended because the use of triggers negatively affects performance.

The format of the timestamp marker variable supplied by Prime Access Registrar is *YYYYMMDDHH24MMSS*. For example, a timestamp of 20131107211050 represents 21:10:50, November 07, 2013.

- **RawAcctRecord**—Used to insert the entire accounting record into the database as a single text field. Contents of this will be whatever is sent by the NAS in the accounting packet and the format is *name=value* pairs delimited with the string specified in Delimiter property. If the delimiter property is not set, the default delimiter is a new line character. RawAcctRecord can be used with the other marker variables.

If multivalued attributes are specified in the marker list, the multiple values are concatenated together with delimiters, and the resulting value will be passed to the insert statement. This delimiter can be specified using the ODBCEnvironmentMultiValueDelimiter property under **/Radius/Advanced**.

## Configuration Examples

This section provides common Oracle accounting configuration examples most likely to be used.

This section contains the following topics:

- [Inserting Selected Attributes into Separate Columns](#)
- [Inserting Complete Accounting Packets into One Column](#)
- [Inserting Selected Attributes into One Column](#)
- [Updating Selected Attributes](#)
- [Deleting Selected Attributes](#)

### Inserting Selected Attributes into Separate Columns

Use the following SQL and MarkerList properties statement to insert selected attributes into separate Oracle columns. The Oracle table definition will have separate columns for each attribute.

```
SQL: "insert into ar_acct (username,nasinfo,packet_type,timestamp) values (?, ?, ?, ?)"
MarkerList: "UserName/SQL_CHAR NAS-Identifier/SQL_CHAR Acct-Status-Type/SQL_CHAR
TimeStamp/SQL_TIMESTAMP"
```

In this example, all the column data types are CHAR/VARCHAR except the timestamp which is DATE. If packet buffering option is disabled, instead of TimeStamp marker, you can also use Oracle's **sysdate** as a value for the timestamp column. The insert statement will look like the following:

```
"insert into ar_acct (username,nasinfo,packet_type,timestamp) values (?,?=?,sysdate)"
```

### Inserting Complete Accounting Packets into One Column

Use SQL and MarkerList properties in the SQLStatement like the following to insert the complete accounting packet into one Oracle column.

```
SQL: "insert into ar_acct (timestamp,raw_packet) values (?,?)"
MarkerList: "TimeStamp/SQL_TIMESTAMP RawAcctRecord/SQL_VARCHAR"
```

### Inserting Selected Attributes into One Column

To insert selected attribute values into one Oracle column delimited by a comma (,), you must configure the AttributeList and Delimiter properties of the odbc-accounting RemoteServer object like the following:

```
AttributeList = "NAS-Identifier,NAS-Port,Acct-Status-Type,Acct-Session-Id"
Delimiter = ,
```

The SQL and MarkerList properties in the SQLStatement will look like the following:

```
SQL: "insert into ar_acct (username,timestamp,attributes) values (?,?=?,)"
MarkerList: "UserName/SQL_CHAR TimeStamp/SQL_TIMESTAMP AttributeList/SQL_VARCHAR"
```

### Updating Selected Attributes

Use the following SQL and MarkerList properties statement to update the selected attributes:

```
SQL: "update arusers_acct set acct_status_type='stop' where username=? and
acct_status_type=?"
MarkerList: "UserName/SQL_CHAR Acct-Status-Type/SQL_CHAR"
```

### Deleting Selected Attributes

Use the following SQL and MarkerList properties statement to delete the selected attributes:

```
SQL = "delete from arusers_acct where username=?"
MarkerList = UserName/SQL_CHAR
```

## Packet Buffering

You can optionally use packet buffering to improve latency when writing accounting records into the database. To enable packet buffering,

- set the BufferAccountingPackets property in the odbc-accounting remote server to TRUE.

This section contains the following topics:

- [When Using Packet Buffering](#)
- [With Packet Buffering Disabled](#)

## When Using Packet Buffering

When `BufferAccountingPackets` is set to `TRUE`, the Prime Access Registrar server's Accounting-Response is returned as soon as the accounting record is successfully written to the local file. To accomplish the queuing of accounting records to a local file, a variant of the existing session backing store is used.

- **Buffered packets** will be inserted into Oracle by a set of background worker threads. The Prime Access Registrar server tries to insert the buffered packet into Oracle for the number of retries configured in the `NumberOfRetriesForBufferedPacket` property (remote odbc accounting server definition). After the configured number of retries, the buffered packets are discarded from the local file.
- **Incoming packets** will be buffered to local file until the configured `MaximumBufferFileSize` is reached. After this limit is reached, no more packets will be addressed. When the database is offline, this remote server will continue to take incoming packets until `MaximumBufferFileSize` reaches. Prime Access Registrar tries to insert these buffered packets when database becomes available.

When using packet buffering, the Prime Access Registrar server can process more incoming packets and can reduce the bottleneck that could occur if the number of simultaneous incoming packets is large and the number of connections to the database is less.

## With Packet Buffering Disabled

When `BufferAccountingPackets` is set to `FALSE`, Accounting-Response is returned after writing the accounting record into Oracle. Oracle write timing is immediate.

- Incoming packets are acknowledged by the remote server only after completing the write into Oracle.
- When the database is offline, no incoming packets are addressed. A slow database server impacts the packet processing rate.

## Dynamic SQL Feature

Using this feature, you can choose the list of SQL statements and the sequence in which the SQL statements need to be executed during run time. This is done through the usage of scripting points.

The SQL-Sequence variable is provided in the Environment Dictionary and it takes the list of SQL statement names and separates each statement name by a semicolon (;). For example, the SQL statement names 'sql3', 'sql4', and 'sql5' are denoted as `sql3;sql4;sql5;`.

While being processed, the packet will be checked for the status of the SQL-Sequence variable. If the variable is set, the list of SQL statements will be executed in the order specified. Even if one of the SQL statements is not found in the configured list of SQL statements, the packet processing fails.

When configured for packet buffering, the `BackingStore` variable in the Environment Dictionary should have the SQL-Sequence variable in order to buffer the SQL-Sequence variable along with the packet information.

# LDAP Accounting

Previous releases of Prime Access Registrar, supported accessing user data from an LDAP server, but this feature was limited to performing authentication and authorization (AA). You can only write the accounting records to local file or Oracle database or proxy to another RADIUS server.

Prime Access Registrar supports writing accounting records into LDAP server enabling integration between billing systems and LDAP.

- Prime Access Registrar adds a new type of service and remote server called `ldap-accounting` that enables inserting accounting records into LDAP.
- You can write accounting records into LDAP by referring this service in `/Radius/DefaultAccountingService` or in the `Accounting-Service` environment variable.

There is no specified schema structure to use the LDAP accounting feature. You can use your own object class design and configure, insert data using `AttributesToWrite` object in the Prime Access Registrar configuration. The Prime Access Registrar server inserts all configured attributes to write the accounting record into LDAP server. This feature is similar to the existing LDAP feature which performs authentication and authorization.

**Note**

---

Prime Access Registrar supports LDAP version 3 client and LDAP version 3 server.

---

## Configuring LDAP Accounting

To use the `ldap` accounting feature,

- you must configure a service of type `ldap-accounting` under `/Radius/Services`.
- You must also configure at least one remote servers of type `ldap-accounting` under `/Radius/RemoteServers`.

This section contains the following topics:

- [LDAP-Accounting Service](#)
- [LDAP RemoteServers](#)
- [Configuration Examples](#)
- [Configuring the LDAP Service for Accounting](#)
- [Configuring an LDAP-Accounting RemoteServer](#)
- [Setting LDAP-Accounting As Accounting Service](#)

## LDAP-Accounting Service

The following is an example of the LDAP-Accounting service:

```
[ //localhost/Radius/Services/ldap_accounting ]
  Name = ldap_accounting
  Description =
  Type = ldap-accounting
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
```

```

OutageScript~ =
MultipleServersPolicy = Failover
RemoteServers/
  1. accounting_server

```

## LDAP RemoteServers

Create a remote server under **/Radius/RemoteServers**, and set its protocol to ldap-accounting. The following is an example of an LDAP-Accounting RemoteServer's configuration:

```

[ //localhost/Radius/RemoteServers/accounting_server ]
Name = accounting_server
Description =
Protocol = ldap-accounting
Port = 389
ReactivateTimerInterval = 300000
Timeout = 15
HostName =
BindName =
BindPassword =
UseSSL = FALSE
EnableKeepAlive = FALSE
DnPath~ =
EntryName~ = (uid=%s)
ObjectClass =
AttributeList =
Delimiter =
LDAPEnvironmentMultiValueDelimiter =
LimitOutstandingRequests = FALSE
MaxOutstandingRequests = 0
EscapeSpecialCharInUserName = FALSE
DNSLookupAndLDAPRebindInterval =
DataSourceConnections = 1
UseLocalTimeZone = FALSE
AttributesToWrite/

```

Table 3-2 lists the properties of LDAP-Accounting RemoteServer.

**Table 3-2** LDAP-Accounting RemoteServer Properties

Fields	Description
Name	Name of the remote server; this property is mandatory and there is no default.
Description	Optional description of server.
Protocol	Must be set to ldap-accounting .
ReactivateTimerInterval	Mandatory time interval (in milliseconds) to activate an inactive server; defaults to 300000 ms.
Timeout	Mandatory time interval (in seconds) to wait for LADP-write operation to complete; defaults to 15 seconds.
DataSourceConnections	Mandatory number of connections to be established; defaults to 8.
EnableKeepAlive	Required; default is FALSE. This is enabled to send a TCP keepalive to keep the idle connection active.
HostName	Required; the LDAP server's hostname or IP address.

Table 3-2 LDAP-Accounting RemoteServer Properties (continued)

Fields	Description
BindName	Optional; the distinguished name (dn) to use when establishing a connection between the LDAP and RADIUS servers.
BindPassword	Optional; the password associated with the <b>BindName</b> .
DnPath	Required; the path that indicates where in the LDAP database to start the write for user information.
EntryName	Required; this specifies the write entry name Prime Access Registrar uses when inserting the LDAP server for user information. When you configure this property, use the notation "%s" to indicate where the user ID should be inserted. For example, a typical value for this property is "(uid=%s)," which means that when inserting for information about user joe, use the fentry name uid=joe.
UseLocalTimeZone	Optional; the default is FALSE. It determines the timezone of accounting records TimeStamp.
AttributeList	List of comma-separated attribute names.
Delimiter	Character used to separate the values of the attributes given in AttributeList property.
AttributesToWrite	List of inserts to be executed to insert the accounting record.
ObjectClass	Required; list of object classes which are all schemas defined in LDAP server. These schemas define required attributes and allowed attributes for an entry which is inserted from Prime Access Registrar.
LDAPEnvironmentMultiValueDelimiter	Optional; allows you to specify a character that separates multi-valued attribute lists when using ldap-accounting.
LimitOutstandingRequests	Required; the default is FALSE. Prime Access Registrar uses this property in conjunction with the <b>MaxOutstandingRequests</b> property to tune the RADIUS server's use of the LDAP server.  When you set this property to TRUE, the number of outstanding requests for this RemoteServer is limited to the value you specified in <b>MaxOutstandingRequests</b> . When the number of requests exceeds this number, Prime Access Registrar queues the remaining requests, and sends them as soon as the number of outstanding requests drops to this number.
MaxOutstandingRequests	Required when you have set the <b>LimitOutstandingRequests</b> to TRUE. The number you specify, which must be greater than zero, determines the maximum number of outstanding requests allowed for this remote server.
EscapeSpecialCharInUserName	FALSE by default.
UseSSL	A boolean field indicating whether you want Prime Access Registrar to use SSL (Secure Socket Layer) when communicating with this RemoteServer. When you set it to TRUE, be sure to specify the <b>CertificateDBPath</b> field in the <b>Advanced</b> section, and be sure the port you specified for this RemoteServer is the SSL port used by the LDAP server.

AttributeList is to specify the list of attribute names separated with comma. When this 'AttributeList' is given in the 'AttributesToWrite' object, these attribute values will be appended together with delimiter specified in 'Delimiter' property and will be supplied as input to that ldap field name.

Attributes from the Prime Access Registrar environment and request dictionaries can be specified in the 'AttributesToWrite' object. Request dictionary will be looked up first for the attributes. Other than the standard attributes in the Prime Access Registrar dictionaries, two new variables are supported inside the 'AttributesToWrite' object.

They are:

- **TimeStamp**—Used to insert the timestamp into LDAP server from Prime Access Registrar. Specifying this will supply the timestamp of that accounting record as a value to the insert. Time zone of this timestamp will be local if UseLocalTimeZone property is set to TRUE, otherwise GMT. This functionality could also be achieved by employing a trigger on the accounting object class in the server.

The format of the timestamp variable supplied by Prime Access Registrar is *YYYYMMDDHH24MMSS*. For example, a timestamp of 20131107211050 represents 21:10:50, November 07, 2013.

- **RawAcctRecord**—Used to insert the entire accounting record into the database as a single text field. Contents of this will be whatever is sent by the NAS in the accounting packet and the format is name=value pairs delimited with the string specified in Delimiter property. If the delimiter property is not set, the default delimiter is a ',' character. RawAcctRecord can be used with the other variables.

If multivalued attributes are specified in the attribute list, the multiple values are concatenated together with delimiters, and the resulting value will be passed to the insert statement. This delimiter can be specified using the LDAPEnvironmentMultiValueDelimiter property.

## Configuration Examples

This section provides common LDAP accounting configuration examples most likely to be used.

This section contains the following topics:

- [Inserting Selected Attributes into Separate LDAP Field](#)
- [Inserting Complete Accounting Packets into One Field](#)
- [Inserting Selected Attributes into One Field](#)

### Inserting Selected Attributes into Separate LDAP Field

Use the following ObjectClass property and 'AttributesToWrite' object properties statement to insert selected attributes into separate LDAP schema. The LDAP schema definition will have separate fields for each attribute.

```
[//localhost/Radius/RemoteServers/accounting-server/AttributesToWrite ]
  sn = timestamp
  uid = username
```

### Inserting Complete Accounting Packets into One Field

Use ObjectClass and 'AttributesToWrite' object properties in the ldap-accounting remote server like the following to insert the complete accounting packet into one LDAP field.

```
[ //localhost/Radius/RemoteServers/accounting-server/AttributeWrites ]
  sealso = rawacctrecord
```

```
uid = username
```

### Inserting Selected Attributes into One Field

To insert selected attribute values into one LDAP field delimited by a comma (,), you must configure the `AttributeList` and `Delimiter` properties of the `ldap-accounting RemoteServer` object like the following:

```
AttributeList = User-Name,NAS-Port,Acct-Session-Id
Delimiter = ,
AttributeWrites/
telephonenumber = attributelist
uid = username
```

## Configuring the LDAP Service for Accounting

You configure an LDAP-Accounting service under `/Radius/Services`. When you define an LDAP-Accounting service under `/Radius/Services`, you must set its type to `ldap-accounting`.

```
[ //localhost/Radius/Services/AR-LDAP-ACCT ]
  Name = AR-LDAP-ACCT
  Description =
  Type = ldap-accounting
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
  OutageScript~ =
  MultipleServersPolicy = Failover
  Remoteservers/
```

**Table 3-3** LDAP-Accounting Service Properties

Fields	Description
Name	Required; inherited from the upper directory.
Description	An optional description of the service.
Type	Must be set to LDAP for LDAP service.
IncomingScript	Optional.
OutgoingScript	Optional.
OutagePolicy	Required; must be set to <b>AcceptAll</b> , <b>DropPacket</b> , or <b>RejectAll</b> . Default is <b>DropPacket</b> .
OutageScript	Optional. if you set this property to the name of a script, Prime Access Registrar runs it when an outage occurs. This property allows you to create a script that notifies you when the RADIUS server detects a failure.
MultipleServersPolicy	Required; must be set to RoundRobin or defaults to Failover.
RemoteServers	Required; list of one or more remote servers defined under <code>/Radius/Services/LDAP/RemoteServers</code> . These servers must be listed in.

This section contains the following topics:

- [MultipleServersPolicy](#)
- [RemoteServers](#)

## MultipleServersPolicy

Use the MultipleServersPolicy property to configure the LDAP remote servers in RoundRobin mode, or the default Failover mode applies. When set to Failover, Prime Access Registrar directs requests to the first server in the **/Radius/Services/LDAP/RemoteServers** list. If that server should fail or go offline, Prime Access Registrar redirects all requests to the next server in the list. The process continues until Prime Access Registrar locates an online server.

When set to RoundRobin, Prime Access Registrar directs each request to the next server in the RemoteServers list to share the resource load across all listed servers.

## RemoteServers

Use the RemoteServers directory to list one or more remote servers to process access requests. The servers must also be listed in order under **/Radius/RemoteServers**.

The order of the RemoteServers list determines the sequence for directing access requests when MultipleServersPolicy is set to RoundRobin mode. The first server in the list receives all access requests when MultipleServersPolicy is set to Failover mode.

## Configuring an LDAP-Accounting RemoteServer

Use the **aregcmd** command **add** to add LDAP servers under **/Radius/RemoteServers**. You must configure an LDAP RemoteServer object for each RemoteServer object you list under **/Radius/Services/LDAP/RemoteServers**.

The Name, Protocol, Port, HostName, BindName, BindPassword, DnPath, and EntryName properties must be configured to use an LDAP remote server.

**Table 3-4** LDAP Remote Server Properties

Fields	Description
Name	Name of the remote server; this property is mandatory and there is no default.
Description	Optional description of server.
Protocol	Must be set to ldap-accounting.
ReactivateTimerInterval	Mandatory time interval (in milliseconds) to activate an inactive server; defaults to 300000 ms.
Timeout	Mandatory time interval (in seconds) to wait for LDAP-write operation to complete; defaults to 15 seconds
DataSourceConnections	Mandatory number of connections to be established; defaults to 8.
EnableKeepAlive	Mandatory field which is enabled to send a TCP keepalive to keep the idle connection active; defaults to FALSE meaning the option is disabled.
HostName	Required; the LDAP server's hostname or IP address.
BindName	Optional; the distinguished name (dn) to use when establishing a connection between the LDAP and RADIUS servers.
BindPassword	Optional; the password associated with the <b>BindName</b> .
DnPath	Required; the path that indicates where in the LDAP database to start the write for user information.

Table 3-4 LDAP Remote Server Properties (continued)

Fields	Description
EntryName	Required; this specifies the write entry name Prime Access Registrar uses when inserting the LDAP server for user information. When you configure this property, use the notation "%s" to indicate where the user ID should be inserted. For example, a typical value for this property is "(uid=%s)," which means that when inserting for information about user joe, use the entry name uid=joe.
UseLocalTimeZone	Set to TRUE or FALSE, determines the timezone of accounting records' TimeStamp (defaults to FALSE).
AttributeList	List of comma-separated attribute names.
Delimiter	Character used to separate the values of the attributes given in AttributeList property.
AttributesToWrite	List of inserts to be executed to insert the accounting record.
ObjectClass	Required; list of object classes which are all schemas defined in LDAP server. These schemas define required attributes and allowed attributes for an entry which is inserted from Prime Access Registrar.
LDAPEnvironmentMultiValueDelimiter	Optional; allows you to specify a character that separates multi-valued attribute lists when using ldap-accounting.
LimitOutstandingRequests	Required; the default is FALSE. Prime Access Registrar uses this property in conjunction with the <b>MaxOutstandingRequests</b> property to tune the RADIUS server's use of the LDAP server.  When you set this property to TRUE, the number of outstanding requests for this RemoteServer is limited to the value you specified in <b>MaxOutstandingRequests</b> . When the number of requests exceeds this number, Prime Access Registrar queues the remaining requests, and sends them as soon as the number of outstanding requests drops to this number.
MaxOutstandingRequests	Required when you have set the <b>LimitOutstandingRequests</b> to TRUE. The number you specify, which must be greater than zero, determines the maximum number of outstanding requests allowed for this remote server.
EscapeSpecialCharInUserName	FALSE by default.
UseSSL	A boolean field indicating whether you want Prime Access Registrar to use SSL (Secure Socket Layer) when communicating with this RemoteServer. When you set it to TRUE, be sure to specify the <b>CertificateDBPath</b> field in the <b>Advanced</b> section, and be sure the port you specified for this RemoteServer is the SSL port used by the LDAP server.

### DNS Look Up and LDAP Rebind Interval

Prime Access Registrar provides a DNS Look-up and LDAP Rebind feature that enables you to use a smart DNS server for LDAP hostname resolution, allows you to query a DNS server at set intervals to resolve the LDAP hostname, and optionally rebind to the LDAP server, if necessary.

When you configure Prime Access Registrar to use an LDAP directory server, you can specify the hostname of the LDAP directory server. The hostname can be a qualified or an unqualified name. You can also specify a timeout period after which Prime Access Registrar will again resolve the hostname. If the IP address returned is different from the previous, Prime Access Registrar establishes a new LDAP bind connection.

The `DNSLookupAndLDAPRebindInterval` property specifies the timeout period after which the Prime Access Registrar server will attempt to resolve the LDAP hostname to IP address (DNS resolution). When you do not modify `DNSLookupAndLDAPRebindInterval`, the default value zero indicates the server will perform normal connection and binding only at start-up time or during a reload. Unless you change the default to a value greater than zero, the server will not perform periodic DNS lookups.

Prime Access Registrar maintains and uses the existing bind connection until a new one is established to minimize any performance impact during the transfer. Prime Access Registrar ensures that no requests are dropped or lost during the transfer to a new LDAP binding.

Set the `DNSLookupAndLDAPRebindInterval` using a numerical value and the letter H for hours or M for minutes, such as in the following examples:

```
set DNSLookupAndLDAPRebindInterval 15M—performs DNS resolution every 15 minutes
```

**Note**

We recommend that you do not set `DNSLookupAndLDAPRebindInterval` to a value less than 15 minutes to minimize its effect on server performance.

```
set DNSLookupAndLDAPRebindInterval 1h—performs DNS resolution every hour
```

**Configuring the DNS Look-up and LDAP Rebind**

To configure the DNS Look-up and LDAP Rebind:

- Step 1** Log into the Prime Access Registrar server, and use `aregcmd` to navigate to `//localhost/Radius/Remoteservers`. If necessary, add the LDAP server, or change directory to it.

```
cd /Radius/RemoteServers/ldap-serv1/
```

- Step 2** Set the `DNSLookupAndLDAPRebindInterval` property to the interval time desired.

```
set DNSLookupAndLDAPRebindInterval 30 M
```

**LDAP Rebind Failures**

Prime Access Registrar records any name resolution failures, bind successes and failures, and the destination hostname and IP address in the log file. At trace level 3, Prime Access Registrar also logs the time of any new bind connections and the closing of any old bind connections.

If either the name resolution or bind attempt fail, Prime Access Registrar continues using the existing bind connection until the timeout has expired again. If there is no existing bind connection, Prime Access Registrar marks the remote server object as *down*.

**Setting LDAP-Accounting As Accounting Service**

Use `aregcmd` to configure the LDAP-accounting Service as the default accounting service under `/Radius` as in the following:

**set DefaultAccountingService AR-LDAP-ACCT**

## MySQL Support

Prime Access Registrar provides support for MySQL to query user records from a MySQL database using odbc interface and enables you to write accounting records into MySQL database using odbc-accounting. Prime Access Registrar has been tested with MySQL 5.0.90 and MyODBC 3.51.27 (reentrant).

For the Prime Access Registrar server to use MySQL, you must create and configure an ODBCDataSource object of type myodbc and a RemoteServer object set to protocol odbc.

**Note**

For more information about dynamic SQL feature, see [Dynamic SQL Feature, page 3-11](#).

This section contains the following topics:

- [Configuring MySQL](#)
- [Example Configuration](#)

## Configuring MySQL

To configure the Prime Access Registrar server to query records form a MySQL database:

- 
- Step 1** Log into the Prime Access Registrar server and launch **aregcmd**.  
Log in as a user with administrative rights such as user **admin**.
- Step 2** Change directory to the **/Radius/Advanced/ODBCDataSources** and add a new ODBCDataSource.
- ```
cd /Radius/Advanced/ODBCDataSources

add mysql
```
- Step 3** Set the new ODBCDataSource type to myodbc.
- ```
cd mysql

set type myodbc
```
- Step 4** Set the Driver property to the path of the MyODBC library.
- Step 5** Set the UserID property to a valid username for the MyODBC database and provide a valid password for this user.
- Step 6** Provide a DataBase name and the name of the Prime Access Registrar RemoteServer object to associate with the ODBCDataSource.
- Step 7** Change directory to **/Radius/RemoteServers** and add a RemoteServer object to associate with the new ODBCDataSource.
- ```
cd /Radius/RemoteServers

add mysql
```

**Step 8** Change directory to the new RemoteServer and set its protocol to odbc-accounting.

```
cd mysql
set protocol odbc-accounting
```

**Step 9** Set the ODBCDataSource property to the name of the ODBCDataSource to associate with this RemoteServer object.

```
set ODBCDataSource mysql
```

---

## Example Configuration

The following shows an example configuration for a MySQL ODBC data source.

```
[ //localhost/Radius/Advanced/ODBCDataSources/mysql ]
Name = mysql
Type = myodbc
Driver = /tmp/libmyodbc3_r.so
UserID = mysql
Password = <encrypted>
DataBase = test
Server = mysql-a
Port = 3306
```

The following shows an example configuration for a RemoteServer

```
Name = odbc-accounting
Description =
Protocol = odbc-accounting
ReactivateTimerInterval = 300000
Timeout = 15
DataSourceConnections = 8
ODBCDataSource =
KeepAliveTimerInterval = 0
BufferAccountingPackets = TRUE
MaximumBufferFileSize = "10 Megabytes"
NumberOfRetriesForBufferedPacket = 3
BackingStoreEnvironmentVariables =
UseLocalTimeZone = FALSE
AttributeList =
Delimiter =
SQLDefinition/
ODBCToRadiusMappings/
ODBCToEnvironmentMappings/
ODBCToCheckItemMappings/
```

## Proxying Accounting Records

You can configure Prime Access Registrar to store accounting records locally and to proxy the accounting records to a remote RADIUS server thereby maintaining multiple accounting logs.

This section contains the following topics:

- [Configuring the Local Cisco Prime Access Registrar Server](#)

- [Configuring the RemoteServer Object](#)

## Configuring the Local Cisco Prime Access Registrar Server

This type of setup requires you to configure the following on the local Prime Access Registrar server:

- A local accounting service of type file
- A remote accounting service of type radius
- An accounting service of type group
- A RemoteServer object

This section contains the following topics:

- [Configuring the Local Accounting Service](#)
- [Configuring the Remote Accounting Service](#)
- [Configuring the Group Accounting Service](#)

### Configuring the Local Accounting Service

The following example shows the configuration required for a local accounting service. This service must be of type file.

```
[//localhost/Radius/Services/accserv1/ ]
  Name = accserv1
  Description =
  Type = file
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
  OutageScript~ =
  FilenamePrefix = accounting
  MaxFileSize = "10 Megabytes"
  MaxFileAge = "1 Day"
  RolloverSchedule =
  UseLocalTimeZone = FALSE
```

### Configuring the Remote Accounting Service

The following example shows the configuration required for a remote accounting service. This service must be of type *radius*, and the name of the remote server must be listed under the RemoteServers subdirectory.

```
[//localhost/Radius/Services/accserv2/
  Name = accserv2
  Description =
  Type = radius
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
  OutageScript~ =
  MultipleServersPolicy = Failover
  RemoteServers/
    1. RemoteRADIUS
```

## Configuring the Group Accounting Service

The following example shows the configuration required for a grouping accounting service. This service must be of type `group` and the local and remote accounting services, `accserv1` and `accserv2` in the previous examples, should be added under the `GroupServices` subdirectory.

The `CiscoAccounting` service groups these two services. The type property should be set to `group`. The services `accserv1` and `accserv2` should be added under `GroupServices` subdirectory of `CiscoAccounting` service.

```
[//localhost/Radius/Services/GroupAccounting/
  Name = GroupAccounting
  Description =
  Type = group
  IncomingScript~ =
  OutgoingScript~ =
  RolloverSchedule =
  ResultRule = AND
  GroupServices/
    1. accserv1
    2. accserv2
```

Refer to [Service Grouping Feature, page 9-14](#), for more information about the Prime Access Registrar Service Grouping feature.

## Configuring the RemoteServer Object

The following example shows the configuration required for the `RemoteServer` object in the local Prime Access Registrar server.

```
[ //localhost/Radius/RemoteServers ]
  Entries 1 to 1 from 1 total entries
  Current filter: <all>

  RemoteRADIUS/
    Name = RemoteRADIUS
    Description =
    Protocol = radius
    IPAddress = aa.bb.cc.dd
    Port = 1812
    ReactivateTimerInterval = 300000
    SharedSecret = secret
    Vendor =
    IncomingScript~ =
    OutgoingScript~ =
    MaxTries = 3
    InitialTimeout = 2000
    AccountingPort = 1813
    ACKAccounting = TRUE
```

If the `ACKAccounting` property is set to `FALSE`, Prime Access Registrar disregards the accounting acknowledgment and continues with the packet processing rather than waiting for the accounting acknowledgment from the Remote server.

If the `ACKAccounting` property is set to `FALSE`, Prime Access Registrar provides the `SendandForget` option. You can set this option to `TRUE`, to delete the original and proxy requests from the buffer that Prime Access Registrar maintains after sending an accounting response to the client.

The group service, CiscoAccounting in this example, should be defined as the default accounting service for any accounting packets received by the local Prime Access Registrar server, as in the following:

```
set /Radius/DefaultAccountingService CiscoAccounting
```



# Diameter

---

Diameter is a networking protocol which is derived from RADIUS protocol. It is considered to be the next generation Authentication, Authorization, and Accounting (AAA) protocol. This is the other core protocol used in the IP Multimedia Subsystem (IMS) architecture for IMS Entities to exchange AAA related information. Cisco Prime Access Registrar (Prime Access Registrar) supports Diameter Applications based on the Diameter Base Protocol defined in RFC 6733.

Diameter is composed of a base protocol and a set of applications which allows it to extend its services to new access technologies. The base protocol provides basic mechanisms for reliable transport, message delivery, and error handling. Each application is defined by an application identifier and associated with commands. Each command is defined with mandatory Attribute Value Pairs (AVPs) and non-mandatory AVPs including vendor-specific AVPs.

The base protocol must be used in conjunction with a Diameter application. Each application relies on the services of the base protocol to support a specific type of network access.

The following is the list of applications supported by Prime Access Registrar:

- Diameter Network Access Server Application (NASREQ, RFC 4005)
- Diameter Base Accounting (RFC 6733)
- Diameter Extensible Authentication Protocol (EAP) Application (RFC 4072)

This chapter contains the following sections:

- [Diameter with EAP Support, page 4-2](#)
- [Diameter Server Startup Log, page 4-3](#)
- [Diameter Stack Level Messages, page 4-4](#)
- [Configuring Authentication and Authorization for Diameter, page 4-6](#)
- [Configuring the Diameter Application in Prime Access Registrar, page 4-8](#)
- [Writing Diameter Application in Prime Access Registrar, page 4-17](#)
- [Translation Framework for Diameter, page 4-21](#)
- [TLS Support for Diameter, page 4-22](#)
- [Managing Diameter Sessions, page 4-24](#)
- [Blacklisting Support for Diameter Remote Server, page 4-24](#)
- [SCTP Multihoming Support for Diameter Client and Remote Server, page 4-24](#)

# Diameter with EAP Support

The Extensible Authentication Protocol (EAP), is an authentication framework which supports multiple authentication mechanisms. EAP may be used on dedicated links, switched circuits, and wired as well as wireless links. For more information on EAP support in Prime Access Registrar, see [Chapter 5, “Extensible Authentication Protocols.”](#)

Prime Access Registrar supports Diameter EAP application that carries EAP packets between a Network Access Server (NAS) working as an EAP Authenticator and a back-end authentication server. The Diameter EAP application is based on the Diameter Network Access Server Application [NASREQ] and is intended for environments similar to NASREQ.

In the Diameter EAP application, authentication occurs between the EAP client and its home Diameter server. This end-to-end authentication reduces the possibility for fraudulent authentication, such as replay and man-in-the-middle attacks. End-to-end authentication also provides a possibility for mutual authentication, which is not possible with PAP and CHAP in a roaming PPP environment.

This topic contains the following sections:

- [Advertising Application Support, page 4-2](#)
- [Diameter EAP Conversation Flow, page 4-2](#)

## Advertising Application Support

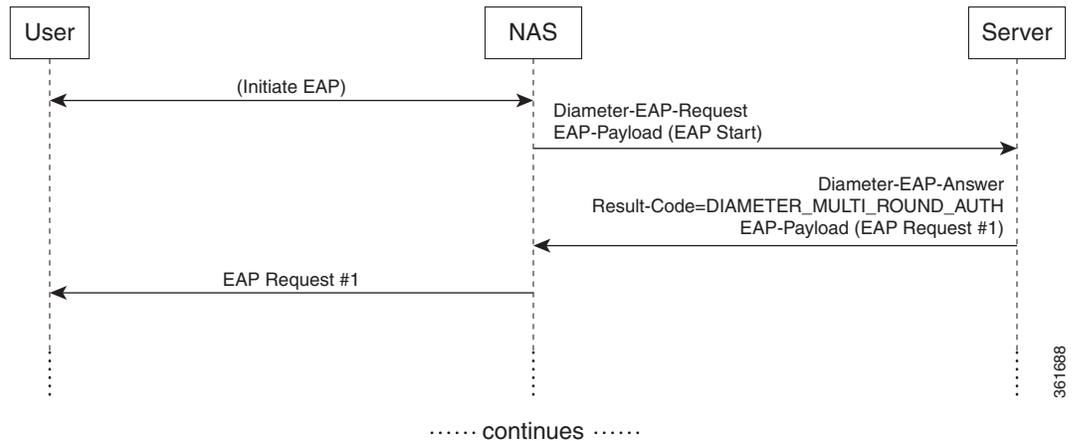
Diameter nodes conforming to this specification must advertise support by including the Diameter EAP Application ID value of 5 in the Auth-Application-Id AVP of the Capabilities-Exchange-Request and Capabilities-Exchange-Answer command [BASE].

If the NAS receives a response with the Result-Code set to `DIAMETER_APPLICATION_UNSUPPORTED` [BASE], it indicates that the Diameter server in the home realm does not support EAP. If possible, the access device may attempt to negotiate another authentication protocol, such as PAP or CHAP. An access device must be cautious when determining whether a less secure authentication protocol will be used, since this could result from a downgrade attack.

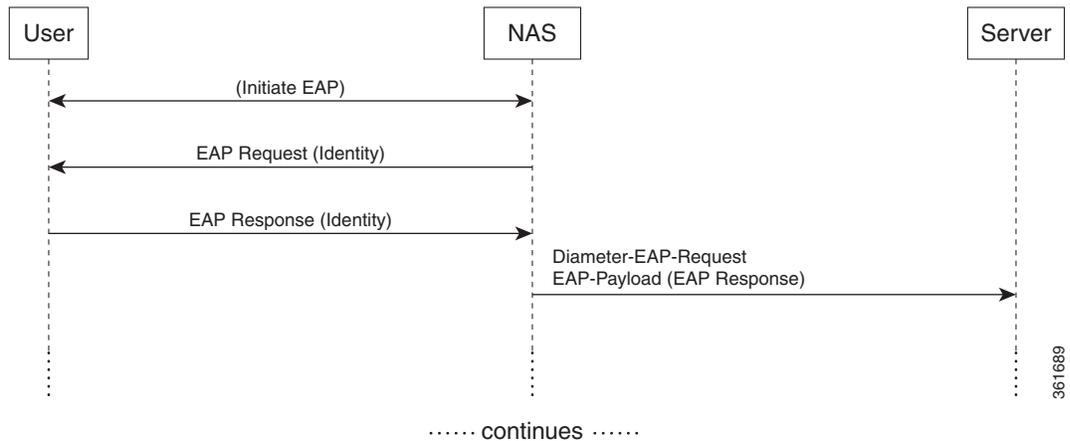
## Diameter EAP Conversation Flow

The EAP conversation between the authenticating peer and the access device begins with the initiation of EAP within a link layer, such as PPP [RFC1661] or IEEE 802.11i [IEEE-802.11i]. Once EAP has been initiated, the access device will typically send a Diameter-EAP-Request message with an empty EAP-Payload AVP to the Diameter server, signifying an EAP-Start. Prime Access Registrar routes the message to the Diameter EAP service through the rules and policy engine (and/or client, server and vendor scripting point) through which the packet is processed. The Diameter EAP Service forms a Diameter-EAP-Answer message containing an EAP-Payload AVP that includes an encapsulated EAP packet. The Result-Code AVP in the message will be set to `DIAMETER_MULTI_ROUND_AUTH`, signifying that a subsequent request is expected.

[Figure 4-1](#) describes the Diameter EAP request flow.

**Figure 4-1** Diameter EAP Request Flow

The access device issues the EAP-Request/Identity message to the EAP client, and forwards the EAP-Response/Identity packet, encapsulated within the EAP-Payload AVP, as a Diameter-EAP-Request to Prime Access Registrar as shown in [Figure 4-2](#). This reduces the number of Diameter message round trips.

**Figure 4-2** Diameter EAP Response Flow

The conversation continues until the Diameter server sends a Diameter-EAP-Answer with a Result-Code AVP indicating success or failure, and an optional EAP-Payload. The Result-Code AVP is used by the access device to determine whether service is to be provided to the EAP client or not. The access device must not rely on the contents of the optional EAP-Payload to determine whether service is to be provided or not.

## Diameter Server Startup Log

When Prime Access Registrar starts, Diameter server also starts.

The log file shows the following:

```

09/30/2013 6:38:47.419 name/radius/1 Info Server 0 Diameter Server Started
09/30/2013 6:38:47.437 name/radius/1 Info Protocol 0 Starting diameter core
  
```

```

09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Product : Cisco Prime
Access Registrar
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Version : 6
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Vendor Id : 0
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Auth Application : 0
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Auth Application : 1
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Acct Application : 3
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Dictionary :
/cisco-ar/conf/diadictionary.xml
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Identity :
10.81.79.43
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Realm : abc.com
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          TCP Listen : 3868
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          SCTP Listen : 3868
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Watch-Dog timeout : 500
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Use IPv6 : 0
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Re-transmission Int : 8
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Max Re-trans Int : 3
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0          Recv Buffer Size : 20480
09/30/2013 6:38:47.448 name/radius/1 Info Protocol 0          Hostnames Used :
10.81.79.43
09/30/2013 6:38:47.448 name/radius/1 Info Protocol 0          Dumping Peer Table
09/30/2013 6:38:47.448 name/radius/1 Info Protocol 0          Expire Time 1
09/30/2013 6:38:47.448 name/radius/1 Info Protocol 0          Peer : Host = 10.77.240.54,
Port = 3868, AdvertiseHostName = , AdvertisedRealm = , TLS = 0
09/30/2013 6:38:47.448 name/radius/1 Info Protocol 0          Peer : Host = 10.77.240.53,
Port = 3868, AdvertiseHostName = , AdvertisedRealm= , TLS = 0
09/30/2013 6:38:47.448 name/radius/1 Info Protocol 0          Dumping Route Table
09/30/2013 6:38:47.448 name/radius/1 Info Protocol 0          Exp Time : 0
09/30/2013 6:38:47.448 name/radius/1 Info Protocol 0          Route : Realm =
dia.com, Action = 2, Redirect-Usage = 0
09/30/2013 6:38:47.448 name/radius/1 Info Protocol 0
Application Id=1, Vendor=0
09/30/2013 6:38:47.449 name/radius/1 Info Protocol 0          Server
= 10.77.240.53, metric = 2
09/30/2013 6:38:47.449 name/radius/1 Info Protocol 0          Auth Stateful Auth : stateful
09/30/2013 6:38:47.449 name/radius/1 Info Protocol 0          Auth Session(T) : 30
09/30/2013 6:38:47.449 name/radius/1 Info Protocol 0          Auth Lifetime(T) : 360
09/30/2013 6:38:47.449 name/radius/1 Info Protocol 0          Auth Grace(T) : 30
09/30/2013 6:38:47.450 name/radius/1 Info Protocol 0          Auth Abort(T) : 20
09/30/2013 6:38:47.450 name/radius/1 Info Protocol 0          Acct Session(T) : 30
09/30/2013 6:38:47.450 name/radius/1 Info Protocol 0          Acct Interim Int : 5
09/30/2013 6:38:47.450 name/radius/1 Info Protocol 0          Acct Real-Time : 0
09/30/2013 6:38:47.450 name/radius/1 Info Protocol 0          Debug Log : enabled
09/30/2013 6:38:47.450 name/radius/1 Info Protocol 0          Trace Log : enabled
09/30/2013 6:38:47.450 name/radius/1 Info Protocol 0          Info Log : enabled
09/30/2013 6:38:47.450 name/radius/1 Info Protocol 0          Console Log : enabled
09/30/2013 6:38:47.450 name/radius/1 Info Protocol 0          Syslog Log : disabled

```

## Diameter Stack Level Messages

The following are the stack level messages that are exchanged between the diameter peers:

- [Capabilities Exchange Message](#)
- [Watchdog Message](#)
- [Disconnect Message](#)

## Capabilities Exchange Message

When Diameter peers establish a transport connection to Prime Access Registrar, they will exchange the Capabilities Exchange messages. This message allows the discovery of a peer's identity and its capabilities (protocol version number, supported Diameter applications, security mechanisms, etc.)

The log file shows the following:

```
05/14/2015 5:36:19.646 name/radius/1 Info Server 0 Starting Server
05/14/2015 5:36:19.707 name/radius/1 Info Server 0 RollingEncryption using new key 17
and aging key 18
05/14/2015 5:36:19.732 name/radius/1 Info Server 0 RollingEncryption using new key 17
and aging key 18
05/14/2015 5:36:19.852 name/radius/1 Info Server 0 Device-Watchdog-Request thread
activated for peer 10.81.79.60
05/14/2015 5:36:19.852 name/radius/1 Info System 0 Connecting to 10.81.79.60:4000
...
05/14/2015 5:36:19.852 name/radius/1 Info System 0 Local socket bind on 10.81.79.81:0
05/14/2015 5:36:20.852 name/radius/1 Info Protocol 0 Connected to RemoteServer b2
05/14/2015 5:36:20.853 name/radius/1 Info Server 0 Initiating CER to 10.81.79.60...
05/14/2015 5:36:20.853 name/radius/1 Info Server 0 Received CEA from the
peer(10.81.79.60), IP: 10.77.240.41
05/14/2015 5:36:20.854 name/radius/1 Info Server 0 Capabilities are successfully
exchanged with 10.81.79.60
05/14/2015 5:36:21.053 name/radius/1 Info Server 0 Sticky Sessions BGwrite thread
activated
05/14/2015 5:36:21.053 name/radius/1 Info Server 0 Sticky Session Count BG thread
activated.
05/14/2015 5:36:21.058 name/radius/1 Info Server 0 Starting Interface 127.0.0.1, port
1812 (RADIUS Access)
05/14/2015 5:36:21.058 name/radius/1 Info Server 0 Starting Interface 127.0.0.1, port
1813 (RADIUS Accounting)
05/14/2015 5:36:21.058 name/radius/1 Info Server 0 Starting Interface 127.0.0.1, port
49 (TACACS+)
05/14/2015 5:36:21.059 name/radius/1 Info Server 0 Starting Interface 127.0.0.1, port
3868 (Diameter-TCP)
05/14/2015 5:36:21.059 name/radius/1 Info Server 0 Starting Interface 127.0.0.1, port
3868 (Diameter-SCTP)
05/14/2015 5:36:21.059 name/radius/1 Info Server 0 Starting Interface 10.81.79.81,
port 1812 (RADIUS Access)
05/14/2015 5:36:21.059 name/radius/1 Info Server 0 Starting Interface 10.81.79.81,
port 1813 (RADIUS Accounting)
05/14/2015 5:36:21.059 name/radius/1 Info Server 0 Starting Interface 10.81.79.81,
port 49 (TACACS+)
05/14/2015 5:36:21.059 name/radius/1 Info Server 0 Starting Interface 10.81.79.81,
port 3868 (Diameter-TCP)
05/14/2015 5:36:21.059 name/radius/1 Info Server 0 Starting Interface 10.81.79.81,
port 3868 (Diameter-SCTP)
05/14/2015 5:36:21.060 name/radius/1 Info Server 0 Starting IPv6 Interface
2001:420:27c1:421:250:56ff:fe99:9e20, port 1812 (RADIUS Access)
05/14/2015 5:36:21.060 name/radius/1 Info Server 0 Starting IPv6 Interface
2001:420:27c1:421:250:56ff:fe99:9e20, port 1813 (RADIUS Accounting)
05/14/2015 5:36:21.060 name/radius/1 Info Server 0 Starting IPv6 Interface
2001:420:27c1:421:250:56ff:fe99:9e20, port 49 (TACACS+)
05/14/2015 5:36:21.060 name/radius/1 Info Server 0 Starting IPv6 Interface
2001:420:27c1:421:250:56ff:fe99:9e20, port 3868 (Diameter-TCP)
05/14/2015 5:36:21.060 name/radius/1 Info Server 0 Starting IPv6 Interface
2001:420:27c1:421:250:56ff:fe99:9e20, port 3868 (Diameter-SCTP)
05/14/2015 5:36:21.060 name/radius/1 Error Configuration 0 Interface
fe80::250:56ff:fe99:9e20: af_bind() to port 1812 failed with -2147418090
05/14/2015 5:36:21.060 name/radius/1 Error Server 0 Failed to start IPv6 Interface
fe80::250:56ff:fe99:9e20, port 1812 (RADIUS Access)
05/14/2015 5:36:21.060 name/radius/1 Info Server 0 Starting Replication Manager
```

```

05/14/2015 5:36:21.060 name/radius/1 Info Server 0 Replication Disabled
05/14/2015 5:36:21.060 name/radius/1 Info Server 0 SNMP is disabled
05/14/2015 5:36:21.061 name/radius/1 Info Server 0 Memory limit for Radius process is
activated
05/14/2015 5:36:21.061 name/radius/1 Info Server 0 Server Started Successfully (pid:
728)

```

## Watchdog Message

The Device-Watchdog-Request and Device-Watchdog-Answer messages are used to proactively detect transport failures. Device Watchdog message time interval is configurable in Prime Access Registrar.

## Disconnect Message

Disconnect messages are initiated when Diameter peers lose transport connection to Prime Access Registrar.

# Configuring Authentication and Authorization for Diameter

This section describes how to configure Prime Access Registrar to perform authentication and authorization and how to configure a local service and userlist.

See for more information on Diameter client properties.

This section contains the following topics:

- [Configuring Local Authentication and Authorization](#)
- [Configuring External Authentication Service](#)

## Configuring Local Authentication and Authorization

In Diameter, an AA-Request packet is a request for authentication and authorization. Authentication checks username and password credentials, while authorization typically involves returning the correct information to allow the service a user is authorized to have. Prime Access Registrar performs AA and returns the appropriate Diameter attributes in an AA-Answer packet.

For adding a Diameter peer in Prime Access Registrar, configure a new entry in the clients (including Policy and Charging Rules Functions (PCRF), Home Subscriber Servers (HSS), Mobility Management Entities (MME), Online Charging Systems (OCS), and others) and remote server object.

The following shows an example configuration for adding a Diameter peer (NAS/Client) in Prime Access Registrar.

```

[ //localhost/Radius/Clients/70dia ]
  Name = 70dia
  Description =
  Protocol = diameter
  HostName = 10.81.79.241
  PeerPort = 3868
  Vendor =
  IncomingScript~ =
  OutgoingScript~ =
  AdvertisedHostName =

```

```

AdvertisedRealm =
InitialTimeout = 1000
MaxIncomingRequestRate = 0
WatchDogTimeout = 500
SCTP-Enabled = FALSE
TLS-Enabled = FALSE

[ //localhost/Radius/Services/diaservice ]
  Name = diaservice
  Description =
  Type = diameter
  IncomingScript~ =
  OutgoingScript~ =
  EnableSticky = FALSE
  MultiplePeersPolicy = Failover
  PeerTimeOutPolicy = FailOver
  DiaRemoteServers/
    Entries 1 to 1 from 1 total entries
    Current filter: <all>

    65/
      Name = 65
      Metric = 2
      Weight = 0
      IsActive = TRUE

```

**Note**

You should restart the Prime Access Registrar server if you change any Diameter specific configuration.

See [and](#) for more details.

## Configuring a Local Service and UserList

See [for](#) more information on how to configure a local service and user list.

The following messages are logged in the trace file at the time of authenticating a valid user:

```

05/14/2015 5:41:47.574: P734: Packet received from 10.81.79.81
05/14/2015 5:41:47.574: P734: Application id: 1, Cmd code: 265, Flag: 0x80
05/14/2015 5:41:47.574: P734: Using Client: vm050
05/14/2015 5:41:47.574: P734: Packet successfully added
05/14/2015 5:41:47.574: P734: Trace of Diameter Packet
05/14/2015 5:41:47.574: P734: Destination-Realm = cisco.com
05/14/2015 5:41:47.574: P734: User-Name = bob
05/14/2015 5:41:47.574: P734: User-Password = <encrypted>
05/14/2015 5:41:47.574: P734: Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015 5:41:47.574: P734: Origin-Host = ar-lnx-vm050.cisco.com
05/14/2015 5:41:47.574: P734: Session-Id = .;2096298391;2
05/14/2015 5:41:47.574: P734: Auth-Application-Id = 1
05/14/2015 5:41:47.574: P734: Origin-Realm = xyz.com
05/14/2015 5:41:47.574: P734: Tracing the packet after running the rules and policies
05/14/2015 5:41:47.574: P734: Using Client: vm050
05/14/2015 5:41:47.574: P734: FastRule Engine called for packet
05/14/2015 5:41:47.574: P734: Fastrule return = 0
05/14/2015 5:41:47.574: P734: Authorizing with Service local-users
05/14/2015 5:41:47.574: P734: Getting User bob's UserRecord from UserList Default
05/14/2015 5:41:47.575: P734: User bob is part of UserGroup PPP-users
05/14/2015 5:41:47.575: P734: Merging UserGroup PPP-users's BaseProfiles into
response dictionary

```

```

05/14/2015 5:41:47.575: P734: Merging UserGroup PPP-users's Attributes into response
Dictionary
05/14/2015 5:41:47.575: P734: Merging attributes into the Response Dictionary:
05/14/2015 5:41:47.575: P734: Trace of Diameter Packet
05/14/2015 5:41:47.575: P734:   User-Name = bob
05/14/2015 5:41:47.575: P734:   Result-Code = Diameter-Success
05/14/2015 5:41:47.575: P734:   Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015 5:41:47.575: P734:   Origin-Host = 10.81.79.81
05/14/2015 5:41:47.575: P734:   Session-Id = .;2096298391;2
05/14/2015 5:41:47.575: P734:   Auth-Application-Id = 1
05/14/2015 5:41:47.575: P734:   Origin-Realm = cisco.com
05/14/2015 5:41:47.575: P734: Sending response to ar-lnx-vm050.cisco.com
05/14/2015 5:41:47.575: P734: Packet successfully removed
05/14/2015 5:41:47.575: P734: Packet Deleted

```

The following messages are logged in the trace file at the time of authenticating an invalid user:

```

05/14/2015 5:45:29.478: P831: Packet received from 10.81.79.81
05/14/2015 5:45:29.478: P831: Application id: 1, Cmd code: 265, Flag: 0x80
05/14/2015 5:45:29.478: P831: Using Client: vm050
05/14/2015 5:45:29.478: P831: Packet successfully added
05/14/2015 5:45:29.478: P831: Trace of Diameter Packet
05/14/2015 5:45:29.478: P831:   Destination-Realm = cisco.com
05/14/2015 5:45:29.478: P831:   User-Name = user.1
05/14/2015 5:45:29.478: P831:   User-Password = <encrypted>
05/14/2015 5:45:29.478: P831:   Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015 5:45:29.479: P831:   Origin-Host = ar-lnx-vm050.cisco.com
05/14/2015 5:45:29.479: P831:   Session-Id = .;2096298391;3
05/14/2015 5:45:29.479: P831:   Auth-Application-Id = 1
05/14/2015 5:45:29.479: P831:   Origin-Realm = xyz.com
05/14/2015 5:45:29.479: P831: Tracing the packet after running the rules and policies
05/14/2015 5:45:29.479: P831: Using Client: vm050
05/14/2015 5:45:29.479: P831: FastRule Engine called for packet
05/14/2015 5:45:29.479: P831: Fastrule return = 0
05/14/2015 5:45:29.479: P831: Authorizing with Service local-users
05/14/2015 5:45:29.479: P831: Getting User user.1's UserRecord from UserList Default
05/14/2015 5:45:29.479: P831: No UserRecord found for User user.1 in UserList
Default, but none _required_ for Authorization.
05/14/2015 5:45:29.479: P831: Trace of Diameter Packet
05/14/2015 5:45:29.479: P831:   User-Name = user.1
05/14/2015 5:45:29.479: P831:   Result-Code = Diameter-Authentication-Rejected
05/14/2015 5:45:29.479: P831:   Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015 5:45:29.479: P831:   Origin-Host = 10.81.79.81
05/14/2015 5:45:29.479: P831:   Session-Id = .;2096298391;3
05/14/2015 5:45:29.479: P831:   Auth-Application-Id = 1
05/14/2015 5:45:29.479: P831:   Origin-Realm = cisco.com
05/14/2015 5:45:29.479: P831: Sending response to ar-lnx-vm050.cisco.com
05/14/2015 5:45:29.479: P831: Packet successfully removed
05/14/2015 5:45:29.480: P831: Packet Deleted

```

## Configuring External Authentication Service

See for more information on how to configure external authentication service.

## Configuring the Diameter Application in Prime Access Registrar

For proxying a diameter application message in Prime Access Registrar, ensure that you do the following:

- [Configuring the Transport Management Properties](#)
- [Registering Applications IDs](#)
- [Configuring the Diameter Peers](#)
- [Configure the Diameter Service](#)

## Configuring the Transport Management Properties

You need to log into the aregcmd using the CLI interface and configure the Transport Management properties in the **Radius/Advanced/Diameter/**.

```
[ //localhost/Radius/Advanced/Diameter ]
  IsDiameterEnabled = TRUE
  General/
    Product = CPAR
    Version = 7.2.0.0
    AuthApplicationIdList = 1
    AcctApplicationIdList = 3
  TransportManagement/
    Identity = 10.77.240.69
    Realm = abc.com
    WatchdogTimeout = 500
    ValidateIncomingMessages = FALSE
    ValidateOutgoingMessages = TRUE
    MaximumNumberOfDiameterPackets = 8194
    ReserveDiameterPacketPool = 0
    DiameterPacketSize = 2048
    AdvertisedHostName/
      SCTPOptions/
        MaxInitRetry = 3
        MaxInboundStream = 4
        MaxOutboundstream = 5
        EnableHeartbeat = FALSE
        HeartbeatInterval = 500
```

You need to set the Identity and AdvertisedHostName properties to IP Address or hostname of the machine in which Prime Access Registrar is installed.

```
--> set Identity 10.77.240.69
Set Identity 10.77.240.69

--> cd AdvertisedHostName
set 1 10.77.240.69
Set the Realm in which Cisco Prime Access Registrar server is present.
--> set Realm cisco.com
Set Realm cisco.com

Save the configuration

--> save

Validating //localhost...
Saving //localhost...

ls
[ //localhost/Radius/Advanced/Diameter/TransportManagement ]
  Identity = 10.77.240.69
  Realm = cisco.com
  WatchdogTimeout = 500
  ValidateIncomingMessages = FALSE
```

```

ValidateOutgoingMessages = TRUE
MaximumNumberOfDiameterPackets = 8194
ReserveDiameterPacketPool = 0
DiameterPacketSize = 2048
AdvertisedHostName/
  1. 10.77.240.69
SCTPOptions/
  MaxInitRetry = 3
  MaxInboundStream = 4
  MaxOutboundstream = 5
  EnableHeartbeat = FALSE
  HeartbeatInterval = 500

```

The description for these properties is available at:

[http://www.cisco.com/en/US/docs/net\\_mgmt/access\\_registrar/5.1/user/guide/objects.html#wp1145662](http://www.cisco.com/en/US/docs/net_mgmt/access_registrar/5.1/user/guide/objects.html#wp1145662)



#### Note

Prime Access Registrar can only listen to one port for diameter connections. In the above configuration, the port number is 3868. All of the diameter clients must use this port number to communicate with the Prime Access Registrar.

## Registering Applications IDs

You need to register the applications IDs for which Prime Access Registrar needs to route the Diameter Messages.

### Registering the Gy application to a diameter stack

To register the Gy application to a diameter stack,

**Step 1** Move to the `//localhost/Radius/Advanced/Diameter/General` directory.

```

[ //localhost/Radius/Advanced/Diameter ]
  IsDiameterEnabled = TRUE
  General/
  TransportManagement/

--> cd General/

[ //localhost/Radius/Advanced/Diameter/General ]
Product = Cisco Prime Access Registrar
Version = 7.2.0.0
AuthApplicationIdList = 1
AcctApplicationIdList =

```

For description of these properties, see .

**Step 2** Set the `AuthApplicationIdList` to list of colon separated values of Application Ids.

```

--> set AuthApplicationIdList "4"

Set AuthApplicationIdList 4

```

## Configuring the Diameter Peers

You need to configure the Diameter Peers such as clients and servers in the `/radius/clients` and `/radius/remoteservers` directories. The following is an example for configuring a Diameter client:

```
[ //localhost/Radius/Clients/70dia ]
Name = 70dia
Description =
Protocol = diameter
HostName = 10.81.79.241
PeerPort = 3868
Vendor =
IncomingScript~ =
OutgoingScript~ =
AdvertisedHostName =
AdvertisedRealm =
InitialTimeout = 1000
MaxIncomingRequestRate = 0
WatchDogTimeout = 500
SCTP-Enabled = FALSE
TLS-Enabled = FALSE
```

The following is an example for configuring a Diameter remote server:

```
[ //localhost/Radius/RemoteServers/lap ]
Name = lap
Description =
Protocol = diameter
HostName = 10.77.144.34
Port = 3868
DestinationRealm = cisco.com
ReactivateTimerInterval = 300000
Vendor =
IncomingScript~ =
OutgoingScript~ =
MaxTries = 3
MaxTPSLimit = 0
MaxSessionLimit = 0
InitialTimeout = 2000
LimitOutstandingRequests = FALSE
MaxPendingPackets = 0
MaxOutstandingRequests = 0
DWatchDogTimeout = 2500
SCTP-Enabled = FALSE
TLS-Enabled = FALSE
AdvertiseHostName =
AdvertiseRealm =
```

For description of these properties, see .



### Note

In order to resolve the hostnames and get the IP addresses, the Prime Access Registrar should either be configured with a DNS server IP, or the client's hostnames and IP addresses should be included in the `/etc/hosts` file.

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 Prime Access Registrar localhost.localdomain localhost
172.16.29.7 GGSN-Gy
::1 localhost6.localdomain6 localhost6
```

## Configure the Diameter Service

To configure the Diameter Service to route the Diameter Messages,

---

**Step 1** Add a Service of type diameter in /Radius/Services/.

```
--> cd /Radius/Services/
--> add dia-proxy

Added dia-proxy

--> cd dia-proxy

[ //localhost/Radius/Services/dia-proxy ]
  Name = dia-proxy
  Description =
  Type =

--> set Type diameter

Set Type diameter
```

**Step 2** Configure the sticky properties.

```
--> set EnableSticky TRUE

Set EnableSticky TRUE

--> ls

[ //localhost/Radius/Services/dia-proxy ]
  Name = dia-proxy
  Description =
  Type = diameter
  IncomingScript~ =
  OutgoingScript~ =
  EnableSticky = TRUE
  StickySessionKey =
  StickyCreationCmdList =
  StickyDeletionCmdList =
  MultiplePeersPolicy = Failover
  PeerTimeOutPolicy = FailOver
  DiaRemoteServers/

--> set StickySessionKey Session-Id#1

Set StickySessionKey Session-Id#1

--> set StickyCreationCmdList 265

Set StickyCreationCmdList 265

--> set StickyDeletionCmdList 275

Set StickyDeletionCmdList 275

--> set MultiplePeersPolicy RoundRobin
```

```

Set MultiplePeersPolicy RoundRobin

--> ls

[ //localhost/Radius/Services/dia-proxy ]
  Name = dia-proxy
  Description =
  Type = diameter
  IncomingScript~ =
  OutgoingScript~ =
  EnableSticky = TRUE
  StickySessionKey = Session-Id#1
  StickyCreationCmdList = 265
  StickyDeletionCmdList = 275
  MultiplePeersPolicy = RoundRobin
  PeerTimeOutPolicy = FailOver
  DiaRemoteServers/

```

**Step 3** Add the peers to which Prime Access Registrar needs to load balance the diameter messages.

```

[ //localhost/Radius/RemoteServers/dial ]
  Name = dial
  Description =
  Protocol = diameter
  HostName = 192.168.30.88
  Port = 3868
  DestinationRealm =
  ReactivateTimerInterval = 300000
  Vendor =
  IncomingScript~ =
  OutgoingScript~ =
  MaxTries = 3
  MaxTPSLimit = 0
  MaxSessionLimit = 0
  InitialTimeout = 2000
  LimitOutstandingRequests = FALSE
  MaxPendingPackets = 0
  MaxOutstandingRequests = 0
  DWatchDogTimeout = 2500
  SCTP-Enabled = FALSE
  TLS-Enabled = FALSE
  AdvertiseHostName =
  AdvertiseRealm =
[ //localhost/Radius/RemoteServers/dia2 ]
  Name = dia2
  Description =
  Protocol = diameter
  HostName =
  Port = 0
  DestinationRealm =
  ReactivateTimerInterval = 300000
  Vendor =
  IncomingScript~ =
  OutgoingScript~ =
  MaxTries = 3
  MaxTPSLimit = 0
  MaxSessionLimit = 0
  InitialTimeout = 2000
  LimitOutstandingRequests = FALSE
  MaxPendingPackets = 0
  MaxOutstandingRequests = 0
  DWatchDogTimeout = 2500
  SCTP-Enabled = FALSE
  TLS-Enabled = FALSE

```

```

    AdvertiseHostName =
    AdvertiseRealm =

--> cd diaRemoteServers/

[ //localhost/Radius/Services/dia-proxy/DiaRemoteServers ]
  Entries 0 to 0 from 0 total entries
  Current filter: <all>

--> add dial

Added dial

--> cd dial/

[ //localhost/Radius/Services/dia-proxy/DiaRemoteServers/dial ]
  Name = dial
  Metric = 2
  Weight = 0
  IsActive = TRUE

--> cd ..

[ //localhost/Radius/Services/dia-proxy/DiaRemoteServers ]
  Entries 1 to 1 from 1 total entries
  Current filter: <all>

  dial/

--> add dia2

Added dia2

--> cd dia2

[ //localhost/Radius/Services/dia-proxy/DiaRemoteServers/dia2 ]
  Name = dia2
  Metric = 3
  Weight = 0
  IsActive = TRUE

```

**Step 4** Save the configuration details.

```

--> save
Validating //localhost...
Saving //localhost...

```

**Step 5** Set DefaultAuthenticationService and DefaultAuthorizationService in /Radius directory.

```

--> set DefaultAuthenticationService dia-proxy

Set DefaultAuthenticationService dia-proxy

--> set DefaultAuthorizationService dia-proxy

Set DefaultAuthorizationService dia-proxy

--> save
Validating //localhost...
Saving //localhost...

--> exit
Logging out of localhost...

```

**Step 6** Restart the Prime Access Registrar server.

```
/cisco-ar/bin/arserver restart
```

The following illustrates the diameter proxy service configuration which load balances the diameter messages to the remote peers.

```
[ /Radius/Services/dia-proxy ]
  Name = dia-proxy
  Description =
  Type = diameter
  IncomingScript~ =
  OutgoingScript~ =
  EnableSticky = TRUE
  StickySessionKey = Session-Id#1
  StickyCreationCmdList = 265
  StickyDeletionCmdList = 275
  MultiplePeersPolicy = RoundRobin
  PeerTimeOutPolicy = FailOver
  DiaRemoteServers/
    Entries 1 to 2 from 2 total entries
    Current filter: <all>

    dial/
      Name = dial
      Metric = 2
      Weight = 0
      IsActive = TRUE
    dia2/
      Name = dia2
      Metric = 3
      Weight = 0
      IsActive = TRUE
```

For description of these properties, see .

## Group-Based Load Balancing in Diameter Proxy Server Configuration

Prime Access Registrar allows you to create two or more groups of Diameter remote servers in a Diameter proxy service configuration. Each of these groups will have a unique set of remote servers, i.e. no two groups will share the same remote server.

The traffic between each of these groups is load-balanced in failover mode; while traffic between remote servers within the same group is load-balanced based on round-robin or failover mode depending on the existing Diameter configuration. The priority of each of the groups is set with the help of metrics.

The workflow for group-based load balancing is as given below:

1. Traffic from Prime Access Registrar to remote server, via Diameter proxy service, is directed through the first group, till Prime Access Registrar has active communication channel with at least one remote server belonging to the first group.
2. When Prime Access Registrar loses connectivity with all the remote servers in the first group, it directs the rest of the Diameter traffic towards remote servers belonging to the second group.
3. Within a group, the load-balancing logic is chosen based on the configuration:
  - a. If the load-balancing logic is configured to be round-robin, the traffic is distributed across all the active remote servers.

- b. If the load-balancing logic is configured to be failover, the traffic is directed towards first priority remote server. When Prime Access Registrar loses connectivity with the first priority remote server, it directs the subsequent traffic towards the second priority remote server. The priority of the Diameter remote servers, in case of failover logic, is set with the help of metrics.

For more information about Diameter server group parameters, see [GroupServers](#), page 2-24.

Following is a sample configuration of the Diameter group server:

```
[ //localhost/Radius/GroupServers ]
  Entries 1 to 2 from 2 total entries
  Current filter: <all>

  Group1/
    Name = Group1
    Description =
    MultiplePeersPolicy = RoundRobin/Failover
    PeerTimeOutPolicy = FailOver/SendError/DropPacket
    DiaRemoteServers/
      Entries 1 to 2 from 2 total entries
      Current filter: <all>

      vm023/
        Name = vm023
        Metric = 1
        Weight = 0
        IsActive = TRUE
      vm045/
        Name = vm045
        Metric = 2
        Weight = 0
        IsActive = TRUE

  Group2/
    Name = Group2
    Description =
    MultiplePeersPolicy = Failover/RoundRobin
    PeerTimeOutPolicy = FailOver/SendError/DropPacket
    DiaRemoteServers/
      Entries 1 to 2 from 2 total entries
      Current filter: <all>

      vm052/
        Name = vm052
        Metric = 1
        Weight = 0
        IsActive = TRUE
      vm062/
        Name = vm062
        Metric = 2
        Weight = 0
        IsActive = TRUE

[ //localhost/Radius/Services/diapro ]
  Name = diapro
  Description =
  Type = diameter
  IncomingScript~ =
  OutgoingScript~ =
  MultiplePeersPolicy = GroupFailover(/Failover/RoundRobin/ImsiRangebased)
  ServerGroups/
    Entries 1 to 2 from 2 total entries
    Current filter: <all>

  Group1/
```

```

        Name = Group1
        Metric = 1
        IsActive = TRUE
Group2/
        Name = Group2
        Metric = 2
        IsActive = TRUE

```

## Writing Diameter Application in Prime Access Registrar

Prime Access Registrar supports extensibility by allowing users to create new:

- authentication/authorization applications
- accounting applications
- command codes
- AVP's

This section contains the following topics:

- [Configuring rex script/service for Diameter](#)
- [Scripting in Diameter](#)
- [Diameter Environment Variables](#)
- [Sample rex script/service](#)
- [Traces/Logs](#)

## Configuring rex script/service for Diameter

To configure script/service for diameter using aregcmd:

- Step 1** Add diameter AVPs in //localhost/Radius/Advanced/DiameterDictionary/DiameterAttributes other than Base stack AVPs.

```

[ //localhost/Radius/Advanced/DiameterDictionary/DiameterAttributes/test ]
  Name = test
  Description =
  Attribute =
  VendorID = 0
  Mandatory = May
  May-Encrypt = No
  Protected = May
  Type =

```

- Step 2** Write a rex script (C/C++) and add it in the scripting point or rex service.

```

[ //localhost/Radius/Services/diaservice ]
  Name = diaservice
  Description =
  Type = rex
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
  OutageScript~ =
  Filename = librexscript.so

```

```
EntryPoint = DiaService
InitEntryPoint =
InitEntryPointArgs =
```

Refer to [Sample rex script/service](#), page 4-19.

---

## Scripting in Diameter

Prime Access Registrar supports 'rex' scripts for Diameter protocol. The script can be configured only as the server incoming script. The commands available for scripting are restricted to 'get' and 'put' on the dictionaries. While setting a value to an attribute, the following convention needs to be followed "<type number>,<value>". For example, if a 'Class' attribute needs to be added to the response dictionary with value as "classvalue", then set it as follows in the script:

```
pResponse->put( pResponse, "Class", "1,classvalue", REX_REPLACE );
```

The following is the list of supported scripting types with the respective type numbers:

```
AVP_STRING_TYPE = 1
AVP_ADDRESS_TYPE = 2
AVP_INTEGER32_TYPE = 3
AVP_UINTEGER32_TYPE = 4
AVP_UTF8_STRING_TYPE = 6
AVP_ENUM_TYPE = 7
AVP_TIME_TYPE = 11
```

Setting response attributes via a script is the only mechanism to add authorization attributes for Diameter requests.

## Diameter Environment Variables

This section lists the environment variables that you can use in scripts for Diameter messages.

[Table 4-1](#) lists the Diameter Environment variables and descriptions.

**Table 4-1** Diameter Environment Variables

| Variable                | Description                                                                                                                                                                                                                                                                         |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Request-Type            | String value.                                                                                                                                                                                                                                                                       |
| Response-Type           | Get/Set the request and response type for diameter packet.<br><b>Sample Values</b><br>Diameter-Access-Request<br>Diameter-Access-Accept<br>Diameter-Access-Reject<br>Diameter-Accounting-Request<br>Diameter-Accounting-Response<br>Diameter-Proxy-Request<br>Diameter-Proxy-Answer |
| Diameter-Application-Id | String value.<br>Get the application id for the packet. For setting in response, need to use Auth-Application-id AVPs.<br><b>Sample Values</b><br>1 (NASREQ)                                                                                                                        |
| Diameter-Command-Code   | String value.<br>Get command code for the diameter packet. It will work only for the access-request packet, not for the accounting request.<br><b>Sample Values</b><br>265 (AA-Request)                                                                                             |

## Sample rex script/service

```
int REXAPI DiaService( int iScriptingPoint,
                      rex_AttributeDictionary_t* pRequest,
                      rex_AttributeDictionary_t* pResponse,
                      rex_EnvironmentDictionary_t* pEnviron )
{
    if( iScriptingPoint == REX_START_SERVICE || iScriptingPoint == REX_STOP_SERVICE )
        return REX_OK;
    int iRetVal = REX_ERROR;
    const char* pszRequestType = pEnviron->get( pEnviron, "Request-Type" );
    const char* pszAppId = pEnviron->get( pEnviron, "Diameter-Application-Id" );
    const char* pszCmdCode = pEnviron->get( pEnviron, "Diameter-Command-Code" );
    if( !( pszRequestType && pszAppId && pszCmdCode ) )
        return iRetVal;
    // check the request type, application id and command code
    /*
Request / Response types
Diameter-Access-Request
Diameter-Access-Accept
Diameter-Access-Reject
Diameter-Accounting-Request
Diameter-Accounting-Response
*/
}
```

```

*/
    if( (strcmp( pszRequestType, "Diameter-Access-Request") == 0) && (strcmp(
pszAppId,"1") ==0 ) && (strcmp( pszCmdCode,"265\
" )== 0 ) )
    {
// our application
// example how to get DiaAttrib from the packet.
    const char* pszSessionId = pRequest ->get( pRequest,"Session-Id",0,0 );
// print in trace
    if( pszSessionId )
        pEnviron->trace( pEnviron, 5, "Diameter Session Id: %s", pszSessionId );
// example: how to add dia attrib in response packet
    pResponse->put( pResponse, "Calling-Station-Id", "1,00-01-02-03-05", REX_APPEND );
    pEnviron->put( pEnviron, "Response-Type", "Diameter-Access-Accept");
    iRetVal = REX_OK;
    }
return iRetVal;
}

```

## Traces/Logs

```

05/14/2015 6:11:05.796: P79: Packet received from 10.81.79.81
05/14/2015 6:11:05.796: P79: Application id: 1, Cmd code: 265, Flag: 0x80
05/14/2015 6:11:05.796: P79: Using Client: vm050
05/14/2015 6:11:05.796: P79: Packet successfully added
05/14/2015 6:11:05.796: P79: Trace of Diameter Packet
05/14/2015 6:11:05.796: P79: Destination-Realm = cisco.com
05/14/2015 6:11:05.796: P79: User-Name = bob
05/14/2015 6:11:05.796: P79: User-Password = <encrypted>
05/14/2015 6:11:05.796: P79: Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015 6:11:05.796: P79: Origin-Host = ar-lnx-vm050.cisco.com
05/14/2015 6:11:05.796: P79: Session-Id = .;2096298391;2
05/14/2015 6:11:05.796: P79: Auth-Application-Id = 1
05/14/2015 6:11:05.796: P79: Origin-Realm = xyz.com
05/14/2015 6:11:05.796: P79: Tracing the packet after running the rules and policies
05/14/2015 6:11:05.796: P79: Using Client: vm050
05/14/2015 6:11:05.796: P79: FastRule Engine called for packet
05/14/2015 6:11:05.796: P79: Fastrule return = 0
05/14/2015 6:11:05.796: P79: Authorizing with Service DiaService
05/14/2015 6:11:05.796: P79: Rex: environ->get( "Request-Type" ) ->
"Diameter-Access-Request"
05/14/2015 6:11:05.797: P79: Rex: environ->get( "Diameter-Application-Id" ) ->
"1"
05/14/2015 6:11:05.797: P79: Rex: environ->get( "Diameter-Command-Code" ) ->
"265"
05/14/2015 6:11:05.797: P79: Rex: request->get( "Session-Id", 0 ) ->
".;2096298391;2"
05/14/2015 6:11:05.797: P79: Diameter Session Id: .;2096298391;2
05/14/2015 6:11:05.797: P79: Rex: response->put( "Calling-Station-Id",
"1,00-01-02-03-05", 0 ) -> TRUE
05/14/2015 6:11:05.797: P79: Rex: environ->put( "Response-Type",
"Diameter-Access-Accept" ) -> TRUE
05/14/2015 6:11:05.797: P79: Trace of Diameter Packet
05/14/2015 6:11:05.797: P79: User-Name = bob
05/14/2015 6:11:05.797: P79: Result-Code = Diameter-Success
05/14/2015 6:11:05.797: P79: Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015 6:11:05.797: P79: Origin-Host = 10.81.79.81
05/14/2015 6:11:05.797: P79: Session-Id = .;2096298391;2
05/14/2015 6:11:05.797: P79: Calling-Station-Id = 1,00-01-02-03-05
05/14/2015 6:11:05.797: P79: Auth-Application-Id = 1
05/14/2015 6:11:05.797: P79: Origin-Realm = cisco.com
05/14/2015 6:11:05.797: P79: Sending response to ar-lnx-vm050.cisco.com

```

```
05/14/2015 6:11:05.797: P79: Packet successfully removed
05/14/2015 6:11:05.797: P79: Packet Deleted
```

## Translation Framework for Diameter

Prime Access Registrar supports translation of an incoming RADIUS request to a Diameter request and vice versa.

The following services are created to set up the translation framework:

- **Radius-Diameter**—For translation of incoming RADIUS request to Diameter equivalent and then the Diameter response to RADIUS equivalent.
- **Diameter-Radius**—For translation of incoming Diameter request to RADIUS equivalent and then the RADIUS response to Diameter equivalent.

For both the translation services, Prime Access Registrar uses the following scripting points to operate on the original packet and on the newly translated packet based on request and response mapping:

- **PreRequestTranslationScript**—To add/modify/delete incoming RADIUS/Diameter attribute values in the request before translation
- **PostRequestTranslationScript**—To add/modify/delete translated Diameter/RADIUS attributes in the request after translation
- **PreResponseTranslationScript**—To add/modify/delete Diameter/RADIUS attribute values in the response before translation
- **PostResponseTranslationScript**—To add/modify/delete RADIUS/Diameter attributes in the response after translation

RADIUS to Diameter translation comes with the 3GPP reverse authorization, if the property is set as True. In that case, the request command mapping must not be defined because the new diameter request is created from the radius request by the 3GPP reverse authorization service. When the diameter response is received from the diameter proxy service, it translates the Diameter response to RADIUS response based on the response mapping configuration and sends radius response to the client.

Prime Access Registrar supports CoA and PoD translation to Re-Auth-Request (RAR) / Abort-Session-Request (ASR), which is triggered directly to Diameter Client without any DRA. Prime Access Registrar sends the translated RAR/ASR packets to client, by configuring a parameter **SendRAR-ASRToClient**. You must also configure the Diameter client to which the packet needs to be sent using the host name of the client in the translation service.

Both these translation services create and maintain appropriate states (with the necessary identifiers, packet pointers, etc) to correlate Request to Response. The states will be cleared if present beyond the 'Timeout' property value and all the retries have been exhausted. You can configure the number of retries under Diameter-RemoteServers.

For more information about the translation parameters, see [Simple Services, page 2-26](#).

### CLI for RADIUS-Diameter Translation

Following is the CLI for RADIUS to Diameter translation:

```
[ //localhost/Radius/Services/rad-dia-trans ]
Name = rad-dia-trans
Description =
Type = radius-diameter
SendRAR-ASRToClient = true
ClientHostName =
```

```

DiameterApplicationId = 5
ProxyServiceName = dia
EnableRequestCommandMappings = true
PreRequestTranslationScript~ = sm
PostRequestTranslationScript~ =
PreResponseTranslationScript~ = env
PostResponseTranslationScript~ =
RequestMapping/
  CommandMappings/
    Radius-CoA-Request = Re-Auth
    Radius-PoD-Request = Abort-Session
  AVPMappings/
    Calling-Station-Id = Session-Id
  AVPsToBeAdded/
    Re-Auth-Request-Type = AUTHORIZE_AUTHENTICATE
  EnvironmentMappings/
ResponseMapping/
  ResultCodeMappings/
    Diameter-Success = Radius-PoD-ACK
    Diameter-Unable-To-Deliver = Radius-PoD-Nak
  AVPMappings/
  AVPsToBeAdded/
  EnvironmentMappings/

```

### CLI for Diameter-RADIUS Translation

Following is the CLI for Diameter to RADIUS translation:

```

[ /Radius/Services/dia-rad ]
  Name = dia-rad
  Description =
  Type = diameter-radius
  ProxyServiceName = rad-proxy
  PreRequestTranslationScript~ =
  PostRequestTranslationScript~ = dia-rad-addpassword
  PreResponseTranslationScript~ =
  PostResponseTranslationScript~ = diareadwritetest
  RequestMapping/
    CommandMappings/
      AA = Radius-Access-Request
    AVPMappings/
      Origin-Host = NAS-Identifier
      User-Name = User-Name
    AVPsToBeAdded/
      NAS-Port = 1
    EnvironmentMappings/
  ResponseMapping/
    ResultCodeMappings/
      Radius-Access-Accept = Diameter-Success
      Radius-Access-Reject = Diameter-Unable-To-Deliver
    AVPMappings/
    AVPsToBeAdded/
    EnvironmentMappings/

```

## TLS Support for Diameter

Prime Access Registrar supports Transport Level Security (TLS) mechanism for Diameter stack. The system provides an option to enable TLS for Diameter client and Diameter remote server. When the TLS option is disabled, communication is established directly using the transport layer without applying any

encryption. The Diameter TLS feature uses the CiscoSSL libraries, which are available as part of the Prime Access Registrar package.

Following is the CLI configuration of a Diameter client with TLS support:

```
[ /Radius/Clients/vm31 ]
  Name = vm31
  Description =
  Protocol = diameter
  HostName = ar-lnx-vm031.cisco.com
  PeerPort = 3868
  Vendor =
  IncomingScript~ =
  OutgoingScript~ =
  AdvertisedHostName =
  AdvertisedRealm =
  MaxIncomingRequestRate = 0
  WatchDogTimeout = 500
  SCTP-Enabled = FALSE
  TLS-Enabled = TRUE
  TLSoptions/
    PrivateKeyPassword = cisco
    ServerCertificateFile = /opt/CSCOar/pki/cert.pem
    ServerKeyFile = /opt/CSCOar/pki/key.pem
    CACertificateFile = /opt/CSCOar/pki/root-cert.pem
    CACertificatePath =
    PeerVerificationMode = None/Optional/RequireCertificate
    VerificationDepth = 4
    EnableAutoChaining = True
```

Following is the CLI configuration of a Diameter remote server with TLS support:

```
[ /Radius/RemoteServers/vm58 ]
  Name = vm58
  Description =
  Protocol = diameter
  HostName = ar-lnx-vm058.cisco.com
  Port = 4322
  DestinationRealm = cisco.com
  ReactivateTimerInterval = 300000
  Vendor =
  IncomingScript~ =
  OutgoingScript~ =
  MaxTries = 3
  InitialTimeout = 2000
  LimitOutstandingRequests = FALSE
  MaxPendingPackets = 0
  MaxOutstandingRequests = 0
  DWatchDogTimeout = 2500
  SCTP-Enabled = FALSE
  TLS-Enabled = TRUE
  AdvertiseHostName =
  AdvertiseRealm =
  TLSoptions/
    PrivateKeyPassword = cisco
    ServerCertificateFile = /opt/CSCOar/pki/cert.pem
    ServerKeyFile = /opt/CSCOar/pki/key.pem
    CACertificateFile = /opt/CSCOar/pki/root-cert.pem
    CACertificatePath =
    PeerVerificationMode = None/Optional/RequireCertificate
    VerificationDepth = 4
    EnableAutoChaining = True
```

For descriptions of the TLS options, see the [Network Resources, page 2-116](#) section of [Chapter 2, “Using the Graphical User Interface.”](#)

## Managing Diameter Sessions

Diameter provides two kinds of services namely authentication/authorization and accounting only (optional). Diameter sessions can be created when an authentication/authorization request comes from an access point or when an accounting start comes from an access point. When a Diameter client issues an authentication request, Prime Access Registrar sends the packet with a Session-Id AVP, which can be used to correlate a Diameter message with a user-session. When a Session Termination Request (STR) message is received from the Diameter client, Prime Access Registrar releases the sessions. Also Re-authentication requests must be mapped to the corresponding user session. In case of accounting packets, the session is created when the accounting start is received from the Diameter client. The session is deleted when the accounting stop message is received.

Prime Access Registrar creates a new session when it receives an authentication or accounting request packet from a Diameter client and when a user session is not already present. It allocates the resources for the particular session from the resource manager and stores the session in a session backing store. This session backing store is a file where session information is written. When a session termination message or an accounting stop message comes from the Diameter client, the session data is deleted from the backing store. Apart from this, Prime Access Registrar maintains the session state for every session it creates. Session cache will be supported for grouped AVPs.

For more information on session manager and its support for Diameter client, see [SessionManagers, page 2-104](#).

## Blacklisting Support for Diameter Remote Server

Prime Access Registrar supports blacklisting of IMSI or IP address values for Diameter remote servers.

You can choose to configure blacklisting as part of the outgoing script of a Diameter remote server with EAP-SIM or EAP-AKA service. For more information about blacklisting, see .

## SCTP Multihoming Support for Diameter Client and Remote Server

Stream Control Transmission Protocol (SCTP) is an IP transport protocol that supports data exchange between exactly two endpoints. Multihoming feature of SCTP provides the ability for a single SCTP endpoint to support multiple IP addresses. With this feature, each of the two endpoints during an SCTP association can specify multiple points of attachment. Each endpoint will be able to receive messages from any of the addresses associated with the other endpoint. With the use of multiple interfaces, data can be sent to alternate addresses when failures occur and thus Prime Access Registrar runs successfully even during network failures.

Prime Access Registrar provides SCTP multihoming support for Diameter client and remote server. With this feature, you can configure multiple source and destination addresses on the Diameter client and remote server.

**Note**

When you use Prime Access Registrar with CentOS, ensure that you configure the Diameter SCTP client and remote servers with different source ports in Prime Access Registrar.

The following shows an example configuration of Diameter remote server with multiple source and destination addresses:

```
[ //localhost/Radius/RemoteServers/Diameter-SCTP-Remote-Server ]
  Name = Diameter-SCTP-Remote-Server
  Description =
  Protocol = diameter
  HostName = 10.197.66.73
  DestinationPort = 3868
  DestinationRealm = cisco.com
  ReactivateTimerInterval = 2000
  Vendor =
  IncomingScript~ =
  OutgoingScript~ =
  MaxTries = 1
  MaxTPSLimit = 0
  MaxSessionLimit = 0
  InitialTimeout = 1500
  LimitOutstandingRequests = FALSE
  MaxPendingPackets = 0
  MaxOutstandingRequests = 0
  DWatchDogTimeout = 2000
  SCTP-Enabled = TRUE
  TLS-Enabled = FALSE
  AdvertiseHostName =
  AdvertiseRealm =
  SCTPParameters/
    SourcePort = 3868
    RTOInitial = 300
    RTOMin = 200
    RTOMax = 300
    MaxInitRetransmits = 8
    AssociationMaxRetrans = 10
    PathMaxRetrans = 10
    RTOCookieLife = 60000
    HBInterval = 50
    SACKTimeout = 400
    InitNumOstreams = 65535
    InitMaxInstreams = 65535
  SCTPAdvertisedHostName/
    Local/
      1. 10.197.66.80
      2. 10.197.66.146
    Remote/
      1. 10.197.66.73
      2. 10.197.66.144
```

The following shows an example configuration of Diameter client with multiple source and destination addresses:

```
[ //localhost/Radius/Clients/Diameter-SCTP-Client ]
  Name = Diameter-SCTP-Client
  Description =
  Protocol = diameter
  HostName = 10.197.66.72
  PeerPort = 3868
  Vendor =
  IncomingScript~ =
```

```
OutgoingScript~ =
AdvertisedHostName =
UserLogEnabled = FALSE
AdvertisedRealm =
InitialTimeout = 1000
MaxIncomingRequestRate = 0
KeepAliveTime = 0
SCTP-Enabled = TRUE
TLS-Enabled = FALSE
SCTPParameters/
  SourcePort = 3868
  RTOInitial = 100
  RTOMin = 100
  RTOMax = 100
  MaxInitRetransmits = 8
  AssociationMaxRetrans = 10
  PathMaxRetrans = 5
  RTOCookieLife = 60000
  HBInterval = 50
  SACKTimeout = 200
  InitNumOstreams = 65535
  InitMaxInstreams = 65535
SCTPAdvertisedHostName/
  Local/
    1. 10.197.66.146
    2. 10.197.66.80
  Remote/
    1. 10.197.66.72
    2. 10.197.66.145
```

For details of the SCTP parameters, see [SCTPParameters Section, page 2-118](#).



## Extensible Authentication Protocols

---

Cisco Prime Access Registrar (Prime Access Registrar) supports the Extensible Authentication Protocol (EAP) to provide a common protocol for differing authentication mechanisms. EAP enables the dynamic selection of the authentication mechanism at authentication time based on information transmitted in the Access-Request. (This type of EAP authentication mechanism is called an authentication exchange.)

Extensible Authentication Protocols (EAP) provide for support of multiple authentication methods. Cisco Prime Access Registrar supports the following EAP authentication methods:

- [EAP-AKA](#)
- [EAP-AKA-Prime \(EAP-AKA'\)](#), page 5-6
- [EAP-FAST](#)
- [EAP-GTC](#)
- [EAP-LEAP](#)
- [EAP-MD5](#)
- [EAP-Negotiate](#)
- [EAP-MSChapV2](#)
- [EAP-SIM](#)
- [EAP-Transport Level Security \(TLS\)](#)
- [EAP-TTLS](#)
- [Protected EAP](#)
  - [PEAP Version 0 \(Microsoft PEAP\)](#)
  - [PEAP Version 1 \(Cisco PEAP\)](#)

In general, you enable each EAP method by creating and configuring a service of the desired type. Use the **radclient** test tool to confirm that the EAP service has been properly configured and is operational.

Both versions of Protected EAP (PEAP) are able to use other EAP methods as the authentication mechanism that is protected by PEAP encryption. For PEAP Version 0, the supported authentication methods are EAP-MSChapV2, EAP-SIM, EAP-TLS and EAP-Negotiate. For PEAP Version 1, the supported authentication methods are EAP-GTC, EAP-SIM, EAP-TLS and EAP-Negotiate.

The PEAP protocol consists of two phases: an authentication handshake phase and a tunnel phase where another complete EAP authentication exchange takes place protected by the session keys negotiated by phase one. Cisco Prime Access Registrar supports the tunneling of other EAP methods within the PEAP phase two exchange.

Prime Access Registrar supports rolling encryption, which involves generating rolling pseudonym secrets for EAP-SIM, EAP-AKA, and EAP-AKA' services. For more details, see [Rolling Encryption Support for Pseudonym Generation in EAP-SIM, EAP-AKA, and EAP-AKA' Services](#), page 5-61.

Prime Access Registrar also supports decryption of encrypted IMSI from the incoming EAP response. For more details, see [Support for Decrypting Encrypted-IMSI for EAP-SIM, EAP-AKA, and EAP-AKA' Services](#), page 5-64.

## EAP-AKA

Authentication and Key Agreement (AKA) is an EAP mechanism for authentication and session key distribution. It is used in the 3rd generation mobile networks Universal Mobile Telecommunications System (UMTS) and CDMA2000. AKA is based on symmetric keys, and typically runs in a UMTS Subscriber Identity Module (USIM), or a (Removable) User Identity Module ((R) UIM), similar to a smart card. EAP-AKA (Extensible Authentication Protocol Method for UMTS Authentication and Key Agreement) includes optional identity privacy support, optional result indications, and an optional fast reauthentication procedure.

In support of EAP-AKA, the following features are supported:

- support of MAP protocol over SIGTRAN
- support of HLR and/or HSS (3GPP compliant)
- Wx interface
- Support M3UA-SIGTRAN over IP

For more information on Wx Interface Support, see the [Wx Interface Support for SubscriberDB Lookup](#), page 9-49.

Prime Access Registrar server supports migration to a converged IP Next Generation Networks (IP NGN) by supporting SS7 and SIGTRAN (SS7 over IP) for HLR communication to enable the seamlessly transition to next-generation IP-based signaling networks.

Prime Access Registrar supports M3UA-SIGTRAN to fetch the authentication vectors from HLR for EAP-AKA authentication, See [SIGTRAN-M3UA](#) for more information.

EAP-AKA is based on rfc-4187 (<http://www.ietf.org/rfc/rfc4187.txt>). This document specifies the details of the algorithms and messages.

This section contains the following topics:

- [Configuring EAP-AKA](#), page 5-2
- [Testing EAP-AKA with radclient](#), page 5-6

## Configuring EAP-AKA

You can use `aregcmd` to create and configure a service of type `eap-aka`.

[Table 5-1](#) lists and describes the EAP-AKA service properties.

**Table 5-1 EAP-AKA Service Properties**

| Property                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AlwaysRequestIdentity    | When True, enables the server to obtain the subscriber's identity via EAP/AKA messages instead of relying on the EAP messages alone. This might be useful in cases where intermediate software layers can modify the identity field of the EAP-Response/Identity message. The default value is False.                                                                                                                                                                                                                                        |
| EnableIdentityPrivacy    | When True, the identity privacy feature is enabled. The default value is False.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| PseudonymSecret          | The secret string that is used as the basis for protecting identities when identity privacy is enabled. This should be at least 16 characters long and have a value that is impossible for an outsider to guess. The default value is secret. This field is not available if the EnableRollingPseudonymSecret field is checked.<br><br><b>Note</b> It is very important to change PseudonymSecret from its default value to a more secure value when identity privacy is enabled for the first time.                                         |
| PseudonymRenewtime       | Specifies the maximum age a pseudonym can have before it is renewed. When the server receives a valid pseudonym that is older than this, it generates a new pseudonym for that subscriber. The value is specified as a string consisting of pairs of numbers and units, where the units might be of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks. The default value is "24 Hours".<br><br>Examples are: "8 Hours", "10 Hours 30 Minutes", "5 D 6 H 10 M"                                                  |
| PseudonymLifetime        | Specifies the maximum age a pseudonym can have before it is rejected by the server, forcing the subscriber to authenticate using its permanent identity. The value is specified as a string consisting of pairs of numbers and units, where the units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks. It can also be Forever, in which case, pseudonyms do not have a maximum age. The default value is "Forever".<br><br>Examples are: "Forever", "3 Days 12 Hours 15 Minutes", "52 Weeks" |
| NotificationService      | (Optional); Notification service is an authorization service and is used to send a notification code to the client in case of an authorization failure. For more information about the Notification-Code variable, see<br><br>This is applicable for RADIUS and Diameter and can be any of the services configured under /radius/services/ except eap services, accounting services, radius-session, radius-query, and diameter.                                                                                                             |
| EnableReauthentication   | When True, the fast reauthentication option is enabled. The default value is False.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| UseOutagePolicyforReauth | Default value is FALSE. When set to TRUE, Prime Access Registrar drops or rejects reauthentication requests as per outage policy when the remote server is down. This can be processed only when there is at least one failed full authentication before proceeding with reauthentication.                                                                                                                                                                                                                                                   |
| MaximumReauthentications | Specifies the maximum number of times a reauthentication identity might be reused before it must be renewed. The default value is 16.                                                                                                                                                                                                                                                                                                                                                                                                        |

Table 5-1 EAP-AKA Service Properties (continued)

| Property                       | Description                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ReauthenticationTimeout        | Specifies the time in seconds that reauthentication identities are cached by the server. Subscribers that attempt to reauthenticate using identities that are older than this value will be forced to use full authentication instead. The default value is 3600 (one hour).                                                                                                         |
| ReauthenticationRealm          | Optional. If you configure the realm, this value is appended to the FastReauthenticationUserId.                                                                                                                                                                                                                                                                                      |
| AuthenticationTimeout          | Time in seconds to wait for authentication to complete. The default is 2 minutes; range is 10 seconds to 10 minutes.                                                                                                                                                                                                                                                                 |
| QuintetGenerationScript~       | Optional. If the script is set, the custom scripting point can be used to read the quintets from a flat file or generate quintets instead of fetching the quintets from HLR. If the script is not set, the Prime Access Registrar sends the request to HLR configured in remote server to fetch the quintets.                                                                        |
| UseProtectedResults            | Enables or disables the use of protected results messages. Results messages indicate the state of the authentication but are cryptographically protected.                                                                                                                                                                                                                            |
| Subscriber_DBLookup            | Required. Must be set to either DIAMETER or SIGTRAN-M3UA.<br><br>When set to DIAMETER, the HSS lookup happens using the Diameter Wx Interface. You need to configure the DestinationRealm to send the Diameter packets to the RemoteServer.<br><br>When set to SIGTRAN-M3UA, the HLR/HSS lookup happens using the SIGTRAN protocol. You need to configure the SIGTRAN remote server. |
| FetchAuthorizationInfo         | Required. When set True, it fetches MSISDN from HLR.<br><br>This field is displayed when you set Subscriber_DBLookup as SIGTRAN-M3UA.                                                                                                                                                                                                                                                |
| IncomingScript~                | Optional script Prime Access Registrar server runs when it receives a request from a client for an EAP-AKA/EAP-SIM service.                                                                                                                                                                                                                                                          |
| OutgoingScript~                | Optional script Prime Access Registrar server runs before it sends a response to a client using an EAP-AKA/EAP-SIM service.                                                                                                                                                                                                                                                          |
| OutageScript~                  | Optional. If set to the name of a script, Prime Access Registrar runs the script when an outage occurs. This property allows you to create a script that notifies you when the server detects a failure.                                                                                                                                                                             |
| RemoteServers                  | Remote server which can provide the service.                                                                                                                                                                                                                                                                                                                                         |
| EnableRollingPseudonymSecret   | Check this box to activate rolling encryption process that involves generating rolling pseudonym secrets for the service.<br><br>For more information about rolling encryption support, see <a href="#">Rolling Encryption Support for Pseudonym Generation in EAP-SIM, EAP-AKA, and EAP-AKA' Services</a> , page 5-61.                                                              |
| Generate3GPPCompliantPseudonym | Optional; the value is set to False by default. If set to TRUE then Prime Access Registrar generates a 12 octet 3GPP compliant pseudonym identity. The Pseudonym username identities are used to protect the privacy of subscriber identities.                                                                                                                                       |

**Table 5-1 EAP-AKA Service Properties (continued)**

| Property             | Description                                                                                                                                                                                                |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Number Of Quintets   | Configured number of authentication vectors from HLR.                                                                                                                                                      |
| SendReAuthIDInAccept | Optional; the value is set to False by default. When set to True, Prime Access Registrar sends SN-Fast-ReAuth-UserName (Starent VSA) in access-accept message.                                             |
| QuintetCacheTimeout  | Required for eap-aka or eap-aka' service; time in seconds an entry remains in the quintet cache. A zero (0) indicates that quintets are not cached. The maximum is 28 days; the default is 0 (no caching). |

To enable EAP-AKA authentication:

---

**Step 1** Launch **aregcmd** and create an EAP-AKA service.

```
cd /Radius/Services
add eap-aka-service
```

**Step 2** Change directory to the service and set its type to eap-aka.

```
cd eap-aka-service
set Type eap-aka
```

---

The following example shows the default configuration for an EAP-AKA service:

```
[ //localhost/Radius/Services/test ]
Name = test
Description =
Type = eap-aka
AlwaysRequestIdentity = False
EnableIdentityPrivacy = False
EnableRollingPseudonymSecret = false
PseudonymSecret = <encrypted>
PseudonymRenewtime = "24 Hours"
PseudonymLifetime = Forever
NotificationService = local-users
Generate3GPPCompliantPseudonym = False
UseOutagePolicyForReauth = False
EnableReauthentication = False
MaximumReauthentications = 16
ReauthenticationTimeout = 3600
ReauthenticationRealm =
AuthenticationTimeout = 120
QuintetGenerationScript~ =
UseProtectedResults = False
SendReAuthIDInAccept = False
SubscriberDBLookup = SIGTRAN-M3UA
FetchAuthorizationInfo = FALSE
MultipleServersPolicy = Failover
IncomingScript~ =
OutgoingScript~ =
OutageScript~ =
RemoteServers/
```

The following example shows the default configuration for an EAP-AKA Wx service:

```
[ //localhost/Radius/Services/eap-aka-wx ]
Name = eap-aka-wx
Description =
Type = eap-aka
AlwaysRequestIdentity = False
EnableIdentityPrivacy = False
PseudonymSecret = <encrypted>
PseudonymRenewtime = "24 Hours"
PseudonymLifetime = Forever
Generate3GPPCompliantPseudonym = False
EnableReauthentication = False
MaximumReauthentications = 16
ReauthenticationTimeout = 3600
ReauthenticationRealm =
AuthenticationTimeout = 120
QuintetGenerationScript~ =
UseProtectedResults = False
SendReAuthIDInAccept = False
SubscriberDBLookup = Diameter
DestinationRealm = mpc.com
PreRequestTranslationScript~ =
PostRequestTranslationScript~ =
PreResponseTranslationScript~ =
PostResponseTranslationScript~ =
```

## Testing EAP-AKA with radclient

To test the EAP-AKA service, launch **radclient** and use the **simple\_eap\_aka\_test** command. The **simple\_eap\_aka\_test** command sends an Access-Request for the designated user with the user's secret key and sequence number.

The response packet should indicate an Access-Accept if authentication was successful. View the response packet to ensure the authentication was successful.

```
simple_eap_aka_test bob secret 2
```

To test from radclient, you have to configure **/cisco-ar/conf/imsi.conf** file on radius server and reload the server. This file content should have imsi users in the format below:

```
<username>:<secret>:<sequence number>
```

For example:

```
bob:bob:1
```

## EAP-AKA-Prime (EAP-AKA')

EAP-AKA-Prime (EAP-AKA') is a new EAP authentication method, with a small revision to the existing EAP-AKA method. EAP-AKA' has a new key derivation function, which binds the keys derived within the method to the name of the access network. This limits the effects of compromised access network nodes and keys.

EAP-AKA' is similar to EAP-AKA in all aspects except the following:

- Key derivation involves an AT\_KDF\_INPUT attribute, which is mapped to the NetworkName attribute, and an AT\_KDF attribute, which takes the key derivation function in the configuration, to ensure that the peer and the server know the name of the access network.
- EAP-AKA' employs SHA-256 (Secure Hash Algorithm) instead of SHA-1 as used in EAP-AKA, to ensure more security.

EAP-AKA' is based on rfc-5448 (<http://www.ietf.org/rfc/rfc5448.txt>). This document specifies the details of the algorithms and messages.

This section contains the following topics:

- [Configuring EAP-AKA'](#), page 5-7
- [Testing EAP-AKA' with radclient](#), page 5-8

## Configuring EAP-AKA'

You can use `aregcmd` to create and configure a service of type `eap-aka-prime`. EAP-AKA' service has the following attribute in addition to the service properties listed in [Table 5-1](#).

| Property    | Description                                                                                                                                                                                                                  |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NetworkName | Required. Name of the access network for which the authentication is performed. This attribute is captured to ensure that the peer and the server know the name of the access network for performing the EAP authentication. |

To enable EAP-AKA' authentication:

- 
- Step 1** Launch `aregcmd` and create an EAP-AKA' service.
- ```
cd /Radius/Services
add eap-aka-prime-service
```
- Step 2** Change directory to the service and set its type to `eap-aka-prime`.
- ```
cd eap-aka-prime-service
set Type eap-aka-prime
```
- 

The following example shows the default configuration for an EAP-AKA' service:

```
//localhost/Radius/Services/eap-aka-prime ]
Name = eap-aka-prime
Description =
Type = eap-aka-prime
AlwaysRequestIdentity = False
EnableIdentityPrivacy = FALSE
EnableRollingPseudonymSecret = false
PseudonymSecret = <encrypted>
PseudonymRenewtime = "24 Hours"
PseudonymLifetime = Forever
NotificationService = local-users
```

```

Generate3GPPCompliantPseudonym = False
EnableReauthentication = FALSE
UseOutagePolicyForReauth = False
MaximumReauthentications = 16
ReauthenticationTimeout = 3600
ReauthenticationRealm =
NetworkName = WAN
AuthenticationTimeout = 120
QuintetGenerationScript~ = aka
UseProtectedResults = TRUE
SendReAuthIDInAccept = False
Subscriber_DBLookup = sigtran
MultipleServersPolicy = Failover
IncomingScript~ =
OutgoingScript~ =
OutageScript~ =
RemoteServers/
    1. sigtran

```

## Testing EAP-AKA' with radclient

To test the EAP-AKA' service, launch **radclient** and use the **simple\_eap\_aka\_prime\_test** command. The **simple\_eap\_aka\_prime\_test** command sends an Access-Request for the designated user with the user's secret key and sequence number.

The response packet should indicate an Access-Accept if authentication was successful. View the response packet to ensure the authentication was successful.

```
simple_eap_aka_prime_test bob secret 2
```

To test from radclient, you have to configure **/cisco-ar/conf/imsi.conf** file on radius server and reload the server. This file content should have imsi users in the format below:

```
<username>:<secret>:<sequence number>
```

For example:

```
bob:bob:1
```

## EAP-FAST

Cisco Prime Access Registrar supports the EAP-FAST authentication method. EAP-FAST uses the EAP-MSChapV2 method for credential provisioning and EAP-GTC for authentication. Credential provisioning typically occurs only during the client's initial EAP-FAST authentication. Subsequent authentications rely on the provisioned credential and will usually omit the provisioning step.

EAP-FAST is an authentication protocol designed to address the performance shortcomings of prior TLS-based EAP methods while retaining features such as identity privacy and support for password-based protocols. The EAP-FAST protocol is described by the IETF draft *draft-cam-winget-eap-fast-00.txt*.

The EAP-FAST credential is known as a Protected Access Credential (PAC) and contains information used to secure the authentication operations. Parts of the PAC are encrypted by the server and are not visible to other entities. Clients are expected to securely store PACs locally for use during authentication.

Configuring EAP-FAST involves creating and configuring the required EAP-MSChapV2 and EAP-GTC services as well as the EAP-FAST service with the appropriate parameters.

You can use the **radclient** test tool to confirm that the EAP services are properly configured and operational.

This section contains the following topics:

- [Configuring EAP-FAST](#)
- [EAP-FAST Keystores](#)
- [Testing EAP-FAST with radclient](#)
- [Parameters Used for Certificate-Based Authentication](#)
- [PAC—Credential Export Utility](#)

## Configuring EAP-FAST

You can use **aregcmd** to create and configure a service of type *eap-fast*.

To enable EAP-FAST:

- 
- Step 1** Launch **aregcmd** and create an EAP-FAST service.
- ```
cd /Radius/Services
add eap-fast-service
```
- Step 2** Change directory to the service and set its type to eap-fast.
- ```
cd eap-fast-service
set type eap-fast
```
- Step 3** Set the AuthorityIdentifier:
- ```
set AuthorityIdentifier authority-identifier
```
- Step 4** : Set the AuthorityInformation:
- ```
set AuthorityInformation authority-information
```
- Step 5** : Set the AuthenticationService:
- ```
set AuthenticationService eap-gtc-service
```
- Step 6** :Set the ProvisionService:
- ```
set ProvisionService eap-mschapv2-service
```
- 

The follow example shows the default configuration for an EAP-FAST service:

```
[ //localhost/Radius/Services/eap-fast-service ]
Name = eap-fast-service
Description =
```

```

Type = eap-fast
IncomingScript~ =
OutgoingScript~ =
MaximumMessageSize = 1024
PrivateKeyPassword = <encrypted>
ServerCertificateFile = /opt/CSCOar/pki/server-cert.pem
ServerKeyFile = /opt/CSCOar/pki/server-key.pem
CACertificateFile = /opt/CSCOar/pki/root-cert.pem
CACertificatePath = /opt/CSCOar/pki
CRLDistributionURL =
ClientVerificationMode = Optional
VerificationDepth = 4
EnableSessionCache = true
UseECCCertificates = true
SessionTimeout = "5 Minutes"
AuthenticationTimeout = 120

```

Table 5-2 lists and describes the EAP-FAST service properties.

**Table 5-2 EAP-FAST Service Properties**

| Property              | Description                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IncomingScript        | Optional script Prime Access Registrar server runs when it receives a request from a client for EAP-FAST service.                                                                                                                                                                                                                                                                |
| OutgoingScript        | Optional script Prime Access Registrar server runs before it sends a response to a client using EAP-FAST.                                                                                                                                                                                                                                                                        |
| AuthorityIdentifier   | A string that uniquely identifies the credential (PAC) issuer. The client uses this value to select the correct PAC to use with a particular server from the set of PACs it might have stored locally.<br><br>Ensure that the AuthorityIdentifier is globally unique and that it does not conflict with identifiers used by other EAP-FAST servers or PAC issuers.               |
| AuthorityInformation  | A string that provides a descriptive text for this credential issuer. The value can be displayed to the client for identification purposes and might contain the enterprise or server names.                                                                                                                                                                                     |
| MaximumMessageSize    | Indicates the maximum length in bytes that a PEAP or EAP-TLS message can have before it is fragmented.                                                                                                                                                                                                                                                                           |
| PrivateKeyPassword    | The password used to protect the server's private key.                                                                                                                                                                                                                                                                                                                           |
| ServerCertificateFile | The full pathname of the file containing the server's certificate or certificate chain used during the TLS exchange. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The two valid encoding prefixes are PEM and DER. If an encoding prefix is not present, the file is assumed to be in PEM format. |

Table 5-2 EAP-FAST Service Properties (continued)

| Property           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ServerKeyFile      | <p>The full pathname of the file containing the server's RSA or ECC private key. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The two valid encoding prefixes are "PEM" and "DER". If an encoding prefix is not present, the file is assumed to be in PEM format.</p> <p>The following example assumes that the subdirectory <b>pki</b> under <b>/cisco-ar</b> contains the server's certificate file. The file <b>server-key.pem</b> is assumed to be in PEM format. The file extension <b>.pem</b> is not significant.</p> <p style="text-align: center;"><b>set ServerKeyFile PEM:/cisco-ar/pki/server-key.pem</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| CACertificateFile  | <p>The full pathname of the file containing trusted CA certificates used for client verification. The file can contain more than one certificate, but all certificates must be in PEM format. DER encoding is not allowed.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| CACertificatePath  | <p>The name of a directory containing trusted CA certificates (in PEM format) used for client verification. This parameter is optional, and if it is used there are some special preparations required for the directory it references.</p> <p>Each certificate file in this directory must contain exactly one certificate in PEM format. The server looks up the certificate files using the MD5 hash value of the certificate's subject name as a key. The directory must therefore also contain a set of symbolic links each of which points to an actual certificate file. The name of each symbolic link is the hash of the subject name of the certificate.</p> <p>For example, if a certificate file named <b>ca-cert.pem</b> is located in the CACertificatePath directory, and the MD5 hash of the subject name contained in <b>ca-cert.path.pem</b> is 1b96dd93, then a symbolic link named 1b96dd93 must point to <b>ca-cert.pem</b>.</p> <p>If there are subject name collisions such as multiple certificates with the same subject name, each link name must be indexed with a numeric extension as in 1b96dd93.0 and 1b96dd93.1.</p> |
| CRLDistributionURL | <p>Optional. Enter the URL that Prime Access Registrar should use to retrieve the CRL. You can specify a URL that uses HTTP or LDAP.</p> <p>The following is an example for an HTTP URL: &lt;</p> <pre>//crl.verisign.com/pca1.1.1.crl&gt;.</pre> <p>The following is an example for an LDAP URL:</p> <pre>ldap://209.165.200.225:388/CN=development-CA,CN=acs-westcoast2,CN=CDP,CN=Public Key Services,CN=Services,CN=Configuration,DC=cisco,DC=com</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

Table 5-2 EAP-FAST Service Properties (continued)

| Property               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ClientVerificationMode | <p>Specifies the type of verification used for client certificates. Must be set to one of RequireCertificate, None, or Optional.</p> <ul style="list-style-type: none"> <li>RequireCertificate causes the server to request a client certificate and authentication fails if the client refuses to provide one.</li> <li>None will not request a client certificate.</li> <li>Optional causes the server to request a client certificate but the client is allowed to refuse to provide one.</li> </ul>                                |
| VerificationDepth      | <p>Specifies the maximum length of the certificate chain used for client verification.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| UseECCCertificates     | <p>Determines the applicability of the authentication mechanism in SmartGrid Solutions, see the <a href="#">Smart Grid Solution Management, page 9-51</a> for more information.</p> <p>When UseECCCertificates is set to True, it can use the ECC, RSA, or combination of both certificate for certificate based verification.</p> <p>When UseECCCertificates is set to False, it can only use the RSA certificate for certificate based verification. The default location to fetch the certificate file is <b>/cisco-ar/pki</b>.</p> |
| EnableSessionCache     | <p>Specifies whether TLS session caching (fast reconnect) is enabled or not. Set to True to enable session caching; otherwise set to False.</p>                                                                                                                                                                                                                                                                                                                                                                                        |
| SessionTimeout         | <p>If TLS session caching (fast reconnect) is enabled, SessionTimeout specifies the maximum lifetime of a TLS session. Expired sessions are removed from the cache and will require a subsequent full authentication.</p> <p>SessionTimeout is specified as a string consisting of pairs of numbers and units, where units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks, as in the following:</p> <p style="text-align: center;"><b>Set SessionTimeout “1 Hour 45 Minutes”</b></p>  |
| AuthenticationTimeout  | <p>Mandatory; specifies time (in seconds) to wait before an authentication request times out; defaults to 120.</p>                                                                                                                                                                                                                                                                                                                                                                                                                     |
| CredentialLifetime     | <p>Specifies the maximum lifetime of a Protected Access Credential (PAC). Clients that successfully authenticate with an expired PAC will be reprovisioned with a new PAC.</p> <p>CredentialLifetime is specified as a string consisting of pairs of numbers and units, where units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks. Credentials that never expire should be specified as Forever.</p>                                                                                 |
| AuthenticationService  | <p>Specifies the name of the EAP-GTC service is used for authentication. The named service must have the UseLabels parameter set to True.</p>                                                                                                                                                                                                                                                                                                                                                                                          |
| ProvisionMode          | <p>Specifies the TLS mode used for provisioning. Clients only support the default Anonymous mode.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                  |

**Table 5-2** EAP-FAST Service Properties (continued)

| Property           | Description                                                                                                                                                                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ProvisionService   | Specifies the name of the EAP-MSChapV2 service used for provisioning.                                                                                                                                                                               |
| AlwaysAuthenticate | Indicates whether provisioning should always automatically rollover into authentication without relying on a separate session. Most environments, particularly wireless, will perform better when this parameter is set to True, the default value. |

**Note**

Prime Access Registrar verifies the certificate during the TLS-based authentication. CRL validation is done before accepting a client certificate during the TLS authentication.

## EAP-FAST Keystores

The EAP-FAST service manages a set of keys used to protect the security and integrity of the PACs it issues. The keys are stored in **/Radius/Advanced/KeyStores/EAP-FAST** and are maintained automatically requiring minimal administration. Administrators can specify the maximum number of keys that are stored and the frequency of key updates.

The following is the default KeyStores settings:

```
[ //localhost/Radius/Advanced/KeyStores/EAP-FAST ]
    NumberOfKeys = 256
    RolloverPeriod = "1 Week"
```

Table 5-3 defines the KeyStores properties.

**Table 5-3** KeyStores Properties

| Property       | Description                                                                         |
|----------------|-------------------------------------------------------------------------------------|
| NumberOfKeys   | Number (from 1-1024) that specifies the maximum number of keys stored for EAP-FAST. |
| RolloverPeriod | Specifies the amount of time between key updates.                                   |

## Testing EAP-FAST with radclient

There are two distinct phases to testing EAP-FAST: provisioning and authentication. In the instructions below, Step 2 and Step 3 test provisioning and Steps 4 and Step 5 test authentication. At least one successful provisioning phase must be completed prior to testing authentication. Testing EAP-FAST with **radclient** requires that the EAP-MSChapV2 and EAP-GTC services be configured and functional.

The following instructions and examples assume that the AlwaysAuthenticate parameter has been set to False for testing purposes. This permits the provisioning and authentication steps to be tested separately. Most installations will set AlwaysAuthenticate to True for production use, and **radclient** works with that setting, but might display extra error messages that you can ignore.

To test EAP-FAST using **radclient**:

---

**Step 1** Start **radclient**.

```
cd /cisco-ar/usrbin
```

```
./radclient -s
```

**Step 2** Specify the inner provisioning method

```
tunnel eap-mschapv2
```

The only allowable method for provisioning is eap-mschapv2.

**Step 3** Provision a new PAC:

```
simple_eap_fast_test user-name password
```

**Step 4** Specify the inner authentication method.

```
tunnel eap-gtc
```

The only allowable method for authentication is eap-gtc.

**Step 5** Authenticate using the PAC.

```
simple_eap_fast_test user-name password
```

---

The **simple\_eap\_fast\_test** command passes its arguments to the inner authentication mechanism which in turn treats the arguments as a username and a password. The command in Step 3 should result in provisioning a new PAC, and Step 5 should result in successful authentication using that PAC.

## PAC Provisioning

The following example provisions a PAC for user bob.

```
pac show
```

```
No PAC(s) available to show
```

```
tunnel eap-mschapv2
```

```
PEAP tunnel method is eap-mschapv2
EAP-FAST tunnel method is eap-mschapv2
```

```
simple_eap_fast_test bob bob
```

```
EAP-FAST authentication status:
  [0x0e07] TLS authentication succeeded
Response to EAP-FAST message was not an Access-Accept
p012
```

```
pac show
```

```
PAC 1 version 1 (219 bytes)
```

```
A-ID      : Prime AR
A-ID-Info : Cisco Prime Access Registrar
I-ID      : bob
Expires   : Never (0)
Key#      : 12
TLV 1     : PAC-Key (1) mandatory (32 bytes)
TLV 2     : PAC-Opaque (2) mandatory (120 bytes)
TLV 3     : PAC-Info (9) mandatory (51 bytes)
```

In this example the **simple\_eap\_fast\_test** command indicates that it did not receive an AccessAccept. This is normal because the provisioning step always results in an AccessReject even when a new PAC has been successfully provisioned. The last **pac show** command displayed some status information from the new PAC and is used to verify that provisioning succeeded and authentication can now be tested. The PAC information displayed will vary and depends on how EAP-FAST is configured.

## Authentication

The following example authenticates user bob (continuing from the [PAC Provisioning](#) example).

### tunnel eap-gtc

```
PEAP tunnel method is eap-gtc
EAP-FAST tunnel method is eap-gtc
```

### simple\_eap\_fast\_test bob bob

```
EAP-FAST authentication status :
[0x0e07] TLS authentication succeeded
SUCCESS : Correctly formatted Session Keys received from the server
p01e
```

In this example, the EAP\_FAST authentication using the PAC from the previous provisioning step succeeded. The AccessAccept packet received from Prime Access Registrar can be displayed to confirm that it contains the expected attributes including the MS-MPPE session keys.

## Parameters Used for Certificate-Based Authentication

EAP-FAST might optionally use RSA or ECC certificates to securely create the tunnel that is used for PAC provisioning. However, the Cisco client does not support the use of certificates and the following parameters will be ignored and should be left at their default values:

- PrivateKeyPassword
- ServerCertificateFile
- ServerKeyFile
- CACertificateFile
- CACertificatePath
- ClientVerificationMode
- VerificationDepth
- UseECCCertificates
- EnableSessionCache

- SessionTimeout

The parameters for configuring certificate-based operation are identical to those used for PEAP and EAP-TLS.

Table 5-4 describes the parameters used for certificate-based authentication.

**Table 5-4 Certificate-Based Authentication Parameters**

| Parameter             | Description                                                                                                                                                                                                                                                                                                                        |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AuthorityIdentifier   | A string that uniquely identifies the credential (PAC) issuer. The client uses this value to select the correct PAC to use with a particular server from the set of PACs it might have stored locally. Care should be taken to ensure that the AuthorityIdentifier is globally unique, that is, is distinct from other PAC issuers |
| AuthorityInformation  | A string that provides some descriptive text for this credential issuer. The value can be displayed to the client for identification purposes. It can contain the enterprise and/or server names.                                                                                                                                  |
| MaximumMessageSize    | Indicates the maximum length in bytes that a EAP-FAST message can have before it is fragmented. If certificates are not used for authentication, fragmentation should not be an issue.                                                                                                                                             |
| AuthenticationTimeout | Indicates the maximum number of seconds before an authentication operation times out and is rejected.                                                                                                                                                                                                                              |
| CredentialLifetime    | Specifies the maximum lifetime of a PAC (Protected Access Credential). Clients that successfully authenticate with an expired PAC will be reprovisioned with a new PAC.                                                                                                                                                            |
| AuthenticationService | Specifies the name of the EAP-GTC service that is used for authentication. The named service must have the UseLabels parameter set to True.                                                                                                                                                                                        |
| ProvisionMode         | Specifies the TLS mode that is used for provisioning. As of this writing, clients only support the default Anonymous mode.                                                                                                                                                                                                         |
| ProvisionService      | Specifies the name of the EAP-MSChapV2 service that is used for provisioning.                                                                                                                                                                                                                                                      |
| AlwaysAuthenticate    | Indicates whether provisioning should always automatically rollover into authentication without relying on a separate session. Most environments, particularly wireless, will perform better when this parameter is set to True (the default value).                                                                               |

## radclient Command Reference

This section describes the **radclient** commands you can use to test EAP-FAST.

### eap-trace

Use the **eap-trace** command to display additional client protocol trace information for EAP methods. Level is a number from 1 to 5 inclusively. Level 5 shows detailed hex dumps of all messages, level 4 shows a message trace without hex dumps, and levels 3 and below show status and error information. To turn off trace displays, set the level to 0.

Set the trace level for all EAP methods.

**eap-trace level**

For example, the following command sets the trace level to 4 for all EAP methods.

```
eap-trace 4
```

Set the trace level for the specified EAP method.

```
eap-trace method level
```

The following example sets the trace level to 5 for EAP-FAST only. The trace level for other EAP methods is not affected.

```
eap-trace eap-fast 5
```



**Note**

The **eap-trace** command is for client-side trace information only and is independent of the server trace level that can be set using **aregcmd**.

## tunnel

The **tunnel** command is used to specify the inner provisioning and authentication methods for EAP-FAST. The specified EAP method type must agree with the server's configured methods or authentication will fail.

```
tunnel eap-method
```

For EAP-FAST provisioning, the only allowable tunnel method is `eap-mchapv2`. For EAP-FAST authentication, the only allowable tunnel method is `eap-gtc`.

## simple\_eap\_fast\_test

The arguments are passed to the inner authentication method as its authentication parameters. If a PAC is not present, the tunnel method should be `eap-mschapv2` and provisioning will occur. If a PAC is present, the tunnel method should be `eap-gtc` and authentication will occur.

```
simple_eap_fast_test username password
```

There are also variants for the **simple** test command for other EAP methods as shown in the following examples:

```
simple_eap_mschapv2_test bob bob
```

```
simple_eap_gtc_test bob bob
```

## pac

The **pac** command is used display, save, and delete PACs that are received from the server during testing. **radclient** maintains a cache of PACs that it knows about and that can be used for authentication testing. The current PAC cache can be displayed with the **pac show** command. PACs created during a test session can be stored to files with the **pac save** command, and reloaded in another session with the **pac load** command. The contents of the PAC cache are completely deleted with **pac delete**. If the optional parameter `cache` is included, PACs are also erased from disk.

```
pac load | save | show { hex } | delete { cache }
```

The **pac show** command displays the currently cached PACs. If the optional parameter *hex* is included, additional detailed information including hex dumps are included in the display output.

```
pac show { hex }
```

The **pac load** command loads any previously saved PACS from disk into the active cache.

The **pac save** command saves all PACs from the active cache to disk. Any previously existing PACS for the same user will be over-written.

The **pac delete** command deletes all PACs from the active cache. If the optional cache parameter is included then PACs are also erased from disk.

```
pac delete { cache }
```

## PAC—Credential Export Utility

You can manually provision EAP-FAST PACs to clients and avoid the use of the protocol provisioning phase. This might be desirable from a security perspective since the default provisioning protocol uses an anonymous (unauthenticated) method to construct the tunnel used to download the PAC to the client.

Manual provisioning involves exporting a PAC from Prime Access Registrar to a file which is then copied to the client machine and used by the import utility. After a PAC has been manually imported, the client should be able to authenticate via EAP-FAST while bypassing the initial provisioning phase. Care should be taken while storing and transporting PAC files since they contain information that potentially allows a client to authenticate via EAP-FAST.

PACs are exported from Prime Access Registrar via the **pac** command which is a new utility for this release. (Note that this **pac** command is a standalone executable which is different from the Radclient **pac** command.) The **pac** command has two capabilities:

- Exports a PAC to a file
- Displays information about an existing PAC file

### PAC Export

Use the **pac export** command to create a new PAC file. In the following example, *eap-fast* is the name of the Prime Access Registrar service configured for EAP-FAST authentication, *bob* is the name of the user this PAC will be used for, and *password* is the password used to derive a key for encrypting the resulting file. (This password is not the same as the administrator's password). The PAC file will be named **bob.pac** by default. You can use the **-f** option to give the file a different name.

```
pac -s export eap-fast bob password
```

If you omit the password parameter, a default password will be used.



#### Note

---

Using the default password is strongly discouraged for security reasons.

---

## PAC Display

Use the **pac show** command to display information about a PAC file. In the following example, **bob.pac** is the name of the PAC file and *password* is the password used to decrypt the file contents.

```
pac -s show bob.pac password
```

## Syntax Summary

The complete **pac** command syntax is as follows:

```
pac { options } export <service-name> <user-name> <file-password>
```

```
pac { options } show <file-name> file-<password>
```

Where:

- C *<cluster>*—Specifies the cluster to be used.
- N *<user>*—Specifies the user.
- P *<user-password>*—Specifies the password to be used.
- s —Logs in using defaults
- v—Enables verbose output
- f—Exports file name (default = {user-name}.pac)

## EAP-GTC

EAP-GTC, defined in RFC 2284, is a simple method for transmitting a user's name and password to an authentication server. EAP-GTC should not be used except as an authentication method for PEAP Version 1 because the password is not protected.

This section contains the following topics:

- [Configuring EAP-GTC](#)
- [Testing EAP-GTC with radclient](#)

## Configuring EAP-GTC

[Table 5-5](#) lists and describes the EAP-GTC specific properties for EAP-GTC authentication.

**Table 5-5** EAP-GTC Properties

| Property    | Description                                                                           |
|-------------|---------------------------------------------------------------------------------------|
| UserService | Required; name of service that can be used to authenticate using cleartext passwords. |

**Table 5-5 EAP-GTC Properties (continued)**

| Property   | Description                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UserPrompt | Optional string the client might display to the user; default is Enter password:” Use the <b>set</b> command to change the prompt, as in the following:<br><br><b>set UserPrompt “Admin Password:”</b> |
| UseLabels  | Required; must be set to TRUE for EAP-FAST authentication and set to FALSE for PEAP authentication. Set to FALSE by default.                                                                           |

To enable EAP-GTC, use **aregcmd** to create and configure a service of type *eap-gtc*

**Step 1** Launch **aregcmd** and create an EAP-GTC service.

```
cd /Radius/Services
add eap-gtc-service
```

**Step 2** Change directory to the service and set its type to eap-gtc.

```
cd eap-gtc-service
set type eap-gtc
```

The follow example shows the default configuration for an EAP-GTC service:

```
[ //localhost/Radius/Services/eap-gtc-service ]
Name = eap-gtc
Description =
Type = eap-gtc
IncomingScript~ =
OutgoingScript~ =
AuthenticationTimeout = 120
UserService =
UserPrompt = "Enter password:"
UseLabels = False
```

**Step 3** Set the service’s UserService to local-users or another local authentication service that is able to authenticate using clear-text passwords.

```
set UserService local-users
```

**Step 4** If configuring for EAP-FAST, set the UseLabels property to TRUE.

## Testing EAP-GTC with radclient

To test the EAP-GTC service, launch **radclient** and use the **simple\_eap\_gtc\_test** command. The **simple\_eap\_gtc\_test** command sends an Access-Request for the designated user with the user’s password.

The response packet should indicate an Access-Accept if authentication was successful. View the response packet to ensure the authentication was successful.

### simple\_eap\_gtc\_test bob bob

```
Packet: code = Access-Accept, id = 2, length = 104, attributes =
  Service-Type = Framed
  Framed-Protocol = PPP
  Framed-IP-Address = 192.168.0.0
  Framed-IP-Netmask = 255.255.255.0
  Framed-Routing = None
  Framed-MTU = 1500
  Framed-Compression = VJ TCP/IP header compression
  Framed-IPX-Network = 1
  EAP-Message = 03:01:00:04
  Ascend-Idle-Limit = 1800
  Message-Authenticator = d3:4e:b1:7e:2d:0a:ed:8f:5f:72:e0:01:b4:ba:c7:e0
```

## EAP-LEAP

Prime Access Registrar supports the new AAA Cisco-proprietary protocol called Light Extensible Authentication Protocol (LEAP), a proprietary Cisco authentication protocol designed for use in IEEE 802.11 wireless local area network (WLAN) environments. Important features of LEAP include:

- Mutual authentication between the network infrastructure and the user
- Secure derivation of random, user-specific cryptographic session keys
- Compatibility with existing and widespread network authentication mechanisms (e.g., RADIUS)
- Computational speed



### Note

Prime Access Registrar supports a subset of EAP to support LEAP. This is not a general implementation of EAP for Prime Access Registrar.

The Cisco-Wireless or Lightweight Extensible Authentication Protocol is an EAP authentication mechanism where the user password is hashed based on an MD4 algorithm and verified by a challenge from both client and server.

## Configuring EAP-LEAP

You can use **aregcmd** to create and configure a service of type **eap-leap**. When you create an EAP-LEAP service type, you must also specify a UserService to perform AAA service. The UserService can be any configured authentication service.

To enable EAP-LEAP:

**Step 1** Launch **aregcmd** and create an EAP-LEAP service.

```
cd /Radius/Services
add eap-leap-service
```

**Step 2** Set the service type to **eap-leap**.

```
cd eap-leap-service
```

```
set type eap-leap
```

```
[ //localhost/Radius/Services/eap-leap-service ]
  Name = newone
  Description =
  Type =
  IncomingScript~ =
  OutgoingScript~ =
  AuthenticationTimeout = 120
  UserService =
```

**Step 3** Set the UserService property to a configured authentication service.

---

## EAP-MD5

Cisco Prime Access Registrar supports EAP-MD5, or MD5-Challenge, another EAP authentication exchange. In EAP-MD5 there is a CHAP-like exchange and the password is hashed by a challenge from both client and server to verify the password is correct. After verified correct, the connection proceeds, although the connection is periodically re-challenged (per RFC 1994).

## Configuring EAP-MD5

Specify type **eap-md5** when you create an EAP-MD5 service. When you create an EAP-MD5 service type, you must also specify a UserService to perform AAA service. The UserService can be any configured authentication service.

You can use **aregcmd** to create and configure a service of type **eap-md5**. When you create an EAP-MD5 service type, you must also specify a UserService to perform AAA service. The UserService can be any configured authentication service.

To enable EAP-MD5:

---

**Step 1** Launch **aregcmd** and create an EAP-LEAP service.

```
cd /Radius/Services
```

```
add eap-md5-service
```

**Step 2** Set the service type to **eap-md5**.

```
cd eap-md5-service
```

```
set type eap-md5
```

```
[ //localhost/Radius/Services/eap-md5-service ]
  Name = newone
  Description =
  Type =
  IncomingScript~ =
```

```
OutgoingScript~ =
AuthenticationTimeout = 120
UserService =
```

**Step 3** Set the UserService property to a configured authentication service.

---

## EAP-Negotiate

EAP-Negotiate is a special service used to select at runtime the EAP service to be used to authenticate the client. EAP-Negotiate is configured with a list of candidate EAP services that represent the allowable authentication methods in preference order. When an EAP session begins, the EAP-Negotiate service tries the first service in the list. If the client does not support that method, it will respond with an EAP-Nak message which triggers EAP-Negotiate to try the next method on the list until a valid method is found or the list is exhausted in which case authentication fails.

EAP-Negotiate is useful when the client population has deployed a mix of different EAP methods that must be simultaneously supported by Prime Access Registrar. It can be difficult or impossible to reliably distinguish which clients require which methods simply by examining RADIUS attributes or other packet properties. EAP-Negotiate solves this problem by using the method negotiation feature of the EAP protocol. Negotiation can be used to select the primary EAP method used for authentication and also to select the inner method for PEAP.

This section contains the following topics:

- [Configuring EAP-Negotiate](#)
- [Negotiating PEAP Tunnel Services](#)
- [Testing EAP-Negotiate with radclient](#)

## Configuring EAP-Negotiate

You may first use **aregcmd** to create and configure the EAP services that will be used for authentication, then create and configure a service of type eap-negotiate.

To enable EAP-Negotiate:

---

**Step 1** Launch **aregcmd** and create an EAP-LEAP service.

```
cd /Radius/Services
add eap-negotiate-service
```

**Step 2** Set the service type to **eap-negotiate**.

```
cd eap-negotiate-service
set type eap-negotiate

[ //localhost/Radius/Services/negotiate ]
Name = negotiate
Description =
Type = eap-negotiate
IncomingScript~ =
```

```
OutgoingScript~ =
AuthenticationTimeout = 120
ServiceList =
```

**Step 3** Set the ServiceList property to a list of preconfigured EAP authentication services.

The ServiceList property lists the names of the EAP services that can be negotiated with this instance of EAP-Negotiate. The ServiceList property is a space-separated list and must consist of valid EAP service name, *not service types*, in preference order from left to right. Each service and type on the list must be unique; duplicates are not allowed.

```
set ServiceList "eap-leap-service eap-md5-service peap-v1-service"
```

---

## Negotiating PEAP Tunnel Services

EAP-Negotiate can also be used to negotiate the inner tunnel service used for phase two of PEAP-V0 or PEAP-V1. To do this, create and configure a service of type `eap-negotiate`. The ServiceList can only contain services that are legal for the version of PEAP that it is used with. Set the PEAP service's TunnelService parameter to the name of the `eap-negotiate` service.



### Note

Not all supplicants support negotiation of the PEAP inner method. EAP-Negotiate can only be used with supplicants that can use EAP-Nak to reject an unsupported inner method.

---

## Testing EAP-Negotiate with radclient

You can test EAP-Negotiate using the same **radclient** commands used to test the other EAP services. For example, you can use the commands for testing `eap-leap` and `peap-v1`.

## EAP-MSChapV2

EAP-MSChapv2 is based on **draft-kamath-pppext-eap-mschapv2-00.txt**, an informational IETF draft document. EAP-MSChapv2 encapsulates the MSChapV2 protocol (specified by RFC 2759) and can be used either as an independent authentication mechanism or as an inner method for PEAP Version 0 (recommended).

This section contains the following topics:

- [Configuring EAP-MSChapV2](#)
- [Testing EAP-MSChapV2 with radclient](#)

## Configuring EAP-MSChapV2

To enable EAP-MSChapv2, use **aregcmd** to create and configure a service of type `eap-mschapv2`

---

**Step 1** Launch **aregcmd** and create an EAP-MSChapV2 service.

```
cd /Radius/Services
add eap-mschapv2
```

**Note**

This example named the service eap-mschapv2, but you can use any valid name for your service.

**Step 2** Set the service's type to eap-mschapv2.

```
cd eap-mschapv2
set Type eap-mschapv2
```

```
[ //localhost/Radius/Services/eap-mschapv2 ]
Name = eap-mschapv2
Description =
Type = eap-mschapv2
IncomingScript~ =
OutgoingScript~ =
AuthenticationTimeout = 120
UserService =
SystemID =
```

**Step 3** Set the service's UserService to local-users or another local authentication service that is able to authenticate using MSChapV2.

```
set UserService local-users
```

**Step 4** You might (optionally) set a string for System ID that identifies the sender of the MSChapV2 challenge message, as in the following:

```
set SystemID system_ID_string
```

## Testing EAP-MSChapV2 with radclient

To test the EAP-MSChapVersion 2 service using **radclient**:

**Step 1** Launch **radclient**.

**Step 2** Use the **simple\_eap\_mschapv2\_test** command to authenticate using EAP-MSChapV2, as in the following:

```
simple_eap_mschapv2_test bob bob
```

```
p006
```

The **simple\_eap\_mschapv2\_test** command above sends an Access-Request for user bob with the user's password. The response packet should indicate an Access-Accept if authentication was successful.

**Step 3** View the response packet to ensure the authentication was successful.

```
p006
```

```
Packet: code = Access-Accept, id = 4, length = 104, attributes =
```

```

Service-Type = Framed
Framed-Protocol = PPP
Framed-IP-Address = 192.168.0.0
Framed-IP-Netmask = 255.255.255.0
Framed-Routing = None
Framed-MTU = 1500
Framed-Compression = VJ TCP/IP header compression
Framed-IPX-Network = 1
EAP-Message = 03:01:00:04
Ascend-Idle-Limit = 1800
Message-Authenticator = 27:90:7e:20:78:34:43:2e:9d:cd:a8:75:82:53:03:65

```

## EAP-SIM

Cisco Prime Access Registrar supports EAP-SIMv16. In a GSM network a subscriber is issued a *smart card* called the subscriber identity module (SIM) that contains a secret key (Ki) and an International Mobile Subscriber Identity (IMSI). The key (Ki) is also stored in the GSM authentication center located with the Home Location Registry (HLR).

An access point uses the Prime Access Registrar RADIUS server to perform EAP-SIM authentication of mobile clients. Prime Access Registrar must obtain authentication information from the HLR. Prime Access Registrar contacts the MAP gateway that performs the MAP protocol over SS7 to the HLR, see [SIGTRAN-M3UA](#) for more information.

In support of EAP-SIM, the Wx Interface feature will be supported. For more information on Wx Interface Support, see the [Wx Interface Support for SubscriberDB Lookup, page 9-49](#).

## Configuring EAP-SIM

You can use **aregcmd** to create and configure a service of type *eap-sim*.

[Table 5-6](#) lists and describes the EAP-SIM specific properties.

**Table 5-6** EAP-SIM Service Properties

| Property              | Description                                                                                                                                                                                                                                                                                           |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AlwaysRequestIdentity | When True, enables the server to obtain the subscriber's identity via EAP/AKA messages instead of relying on the EAP messages alone. This might be useful in cases where intermediate software layers can modify the identity field of the EAP-Response/Identity message. The default value is False. |
| EnableIdentityPrivacy | When True, the identity privacy feature is enabled. The default value is False.                                                                                                                                                                                                                       |

**Table 5-6 EAP-SIM Service Properties (continued)**

| Property                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PseudonymSecret          | <p>The secret string that is used as the basis for protecting identities when identity privacy is enabled. This should be at least 16 characters long and have a value that is impossible for an outsider to guess. The default value is secret. This field is not available if the EnableRollingPseudonymSecret field is checked.</p> <p><b>Note</b> It is very important to change PseudonymSecret from its default value to a more secure value when identity privacy is enabled for the first time.</p>                                         |
| PseudonymRenewtime       | <p>Specifies the maximum age a pseudonym can have before it is renewed. When the server receives a valid pseudonym that is older than this, it generates a new pseudonym for that subscriber. The value is specified as a string consisting of pairs of numbers and units, where the units might be of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks. The default value is "24 Hours".</p> <p>Examples are: "8 Hours", "10 Hours 30 Minutes", "5 D 6 H 10 M"</p>                                                  |
| PseudonymLifetime        | <p>Specifies the maximum age a pseudonym can have before it is rejected by the server, forcing the subscriber to authenticate using its permanent identity. The value is specified as a string consisting of pairs of numbers and units, where the units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks. It can also be Forever, in which case, pseudonyms do not have a maximum age. The default value is "Forever".</p> <p>Examples are: "Forever", "3 Days 12 Hours 15 Minutes", "52 Weeks"</p> |
| NotificationService      | <p>(Optional); Notification service is an authorization service and is used to send a notification code to the client in case of an authorization failure. For more information about the Notification-Code variable, see</p> <p>This can be any of the services configured under /radius/services/ except eap services, accounting services, radius-session, radius-query, and diameter.</p>                                                                                                                                                       |
| EnableReauthentication   | <p>When True, the fast reauthentication option is enabled. The default value is False.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| UseOutagePolicyforReauth | <p>Default value is FALSE. When set to TRUE, Prime Access Registrar drops or rejects reauthentication requests as per outage policy when the remote server is down. This can be processed only when there is at least one failed full authentication before proceeding with reauthentication.</p>                                                                                                                                                                                                                                                   |
| MaximumReauthentications | <p>Specifies the maximum number of times a reauthentication identity might be reused before it must be renewed. The default value is 16.</p>                                                                                                                                                                                                                                                                                                                                                                                                        |
| ReauthenticationTimeout  | <p>Specifies the time in seconds that reauthentication identities are cached by the server. Subscribers that attempt to reauthenticate using identities that are older than this value will be forced to use full authentication instead. The default value is 3600 (one hour).</p>                                                                                                                                                                                                                                                                 |
| ReauthenticationRealm    | <p>Optional. If you configure the realm, this value is appended to the FastReauthenticationUserId.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                              |

Table 5-6 EAP-SIM Service Properties (continued)

| Property                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AuthenticationTimeout        | Time in seconds to wait for authentication to complete. The default is 2 minutes; range is 10 seconds to 10 minutes.                                                                                                                                                                                                                                                                                                                             |
| QuintetGenerationScript~     | Optional. If the script is set, the custom scripting point can be used to read the quintets from a flat file or generate quintets instead of fetching the quintets from HLR. If the script is not set, the Prime Access Registrar sends the request to HLR configured in remote server to fetch the quintets.                                                                                                                                    |
| UseProtectedResults          | Enables or disables the use of protected results messages. Results messages indicate the state of the authentication but are cryptographically protected.                                                                                                                                                                                                                                                                                        |
| TripletCacheTimeout          | Required; timeout value of triplet cache.                                                                                                                                                                                                                                                                                                                                                                                                        |
| SubscriberDBLookup           | Required. Must be set to either DIAMETER or SIGTRAN-M3UA.<br>When set to DIAMETER, the HSS lookup happens using the Diameter Wx Interface. You need to configure the DestinationRealm to send the Diameter packets to the RemoteServer.<br>When set to SIGTRAN-M3UA, the HLR/HSS lookup happens using the SIGTRAN protocol. You need to configure the SIGTRAN remote server.<br>When set to MAP, the HLR lookup happens using the MAP interface. |
| FetchAuthorizationInfo       | Required. When set True, it fetches MSISDN from HLR.<br><br>This field is displayed when you set Subscriber_DBLookup as SIGTRAN-M3UA.                                                                                                                                                                                                                                                                                                            |
| IncomingScript~              | Optional script Prime Access Registrar server runs when it receives a request from a client for an EAP-AKA/EAP-SIM service.                                                                                                                                                                                                                                                                                                                      |
| OutgoingScript~              | Optional script Prime Access Registrar server runs before it sends a response to a client using an EAP-AKA/EAP-SIM service.                                                                                                                                                                                                                                                                                                                      |
| OutageScript~                | Optional. If set to the name of a script, Prime Access Registrar runs the script when an outage occurs. This property allows you to create a script that notifies you when the server detects a failure.                                                                                                                                                                                                                                         |
| RemoteServers                | Remote server which can provide the service.                                                                                                                                                                                                                                                                                                                                                                                                     |
| EnableRollingPseudonymSecret | Check this box to activate rolling encryption process that involves generating rolling pseudonym secrets for the service.<br><br>For more information about rolling encryption support, see <a href="#">Rolling Encryption Support for Pseudonym Generation in EAP-SIM, EAP-AKA, and EAP-AKA' Services</a> , page 5-61.                                                                                                                          |

To enable EAP-SIM authentication using aregcmd:

**Step 1** Launch **aregcmd** and create an EAP-SIM service.

```
cd /Radius/Services
add eap-sim-service
```

**Step 2** Change directory to the service and set its type to *eap-sim*.

```
cd eap-sim-service
```

```
set Type eap-sim
```

```
[ //localhost/Radius/Services/EAP-SIM ]
  Name = EAP-SIM
  Description =
  Type = eap-sim
  NumberOfTriplets = 2
  UseSimDemoTriplets = False
  AlwaysRequestIdentity = False
  EnableIdentityPrivacy = False
  EnableRollingPseudonymSecret = false
  PseudonymSecret = <encrypted>
  PseudonymRenewtime = "24 Hours"
  PseudonymLifetime = Forever
  NotificationService = local-users
  Generate3GPPCompliantPseudonym = False
  EnableReauthentication = False
  UseOutagePolicyForReauth = False
  MaximumReauthentications = 16
  ReauthenticationTimeout = 3600
  ReauthenticationRealm =
  TripletCacheTimeout = 120
  AuthenticationTimeout = 120
  UseProtectedResults = False
  SendReAuthIDInAccept = False
  SubscriberDBLookup = SIGTRAN-M3UA
  FetchAuthorizationInfo = FALSE
  MultipleServersPolicy = Failover
  IncomingScript~ =
  OutgoingScript~ =
  OutageScript~ =
  RemoteServers/

[ //localhost/Radius/Services/eap-sim-wx ]
Name = eap-sim-wx
Description =
Type = eap-sim
NumberOfTriplets = 2
UseSimDemoTriplets = False
AlwaysRequestIdentity = False
EnableIdentityPrivacy = False
PseudonymSecret = <encrypted>
PseudonymRenewtime = "24 Hours"
PseudonymLifetime = Forever
Generate3GPPCompliantPseudonym = False
EnableReauthentication = False
MaximumReauthentications = 16
ReauthenticationTimeout = 3600
ReauthenticationRealm =
TripletCacheTimeout = 120
AuthenticationTimeout = 120
UseProtectedResults = False
SendReAuthIDInAccept = False
SubscriberDBLookup = DIAMETER
DestinationRealm = hss.com
PreRequestTranslationScript~ =
PostRequestTranslationScript~ =
```

```
PreResponseTranslationScript~ =
PostResponseTranslationScript~
```

**Note**

The EAP-SIM property `OutagePolicy` present in earlier versions of Prime Access Registrar is no longer part of the EAP-SIM configuration.

To enable EAP-SIM authentication using **radclient**:

- Step 1** Create an EAP-SIM service.
- Step 2** Change directory to the service and set its type to `eap-sim`.
- Step 3** Execute the below command in `radclient` to set session keys in the server.

```
simple_eap_sim_test 987456321123654 secret
```

**Note**

The IMSI number that is stored in HLR is used for EAP-SIM authentication.

- Step 4** Enter the server name in which the session key is created to view the `eap-sim` service details.

```
p006
```

```
Packet: code = Access-Accept, id = 3, length = 207, attributes =
User-Name = 987456321123654
MS-MPPE-Send-Key =
9c:56:e5:36:9f:fe:84:a2:26:16:80:0a:13:74:fb:b7:87:30:00:5c:45:99:ea:78:af:7d:ae:37:0e
:b1:3a:2e:2b:b1:c8:4f:20:39:33:04:eb:dc:ba:27:e7:6f:56:08:21:56
EAP-Message = 03:02:00:04
Cisco-AVPair = auth-algo-type=eap-sim
MS-MPPE-Recv-Key =
8b:27:42:c5:47:79:ce:6a:41:ae:34:1f:15:2f:cf:b8:ee:18:e7:b5:1c:64:41:26:f7:4b:bc:53:bd
:54:57:70:a3:3b:df:78:9e:34:33:47:b3:a2:ff:4e:f1:fe:6f:8f:ee:aa
Message-Authenticator = 45:02:01:97:55:3d:bc:80:34:76:a4:5a:6b:29:ac:bc
```

## Quintets to Triplets Conversion

Prime Access Registrar provides a configuration option in EAP-SIM service, which allows conversion of quintets received from a Universal Mobile Telecommunications Service (UMTS) subscriber to triplets. This feature facilitates backward compatibility by allowing to perform EAP-SIM authentication from an EAP-AKA or EAP-AKA' source.

## EAP-Transport Level Security (TLS)

EAP-Transport Level Security (EAP-TLS), described in RFC 2716, is an authentication method designed to mitigate several weaknesses of EAP. EAP-TLS leverages TLS, described in RFC 2246, to achieve certificate-based authentication of the server and (optionally) the client. EAP-TLS provides many of the same benefits as PEAP but differs from it in the lack of support for legacy authentication methods.

This section contains the following topics:

- [Configuring EAP-TLS](#)
- [Configuring EAP-TLS with OCSP Support](#)
- [Testing EAP-TLS with RSA or ECC Certificate using radclient](#)
- [Testing EAP-TLS with Client Certificates](#)

### Configuring EAP-TLS

You can use `aregcmd` to create and configure a service of type `eap-tls`. [Table 5-7](#) describes the EAP-TLS configuration properties:

**Table 5-7 EAP-TLS Service Properties**

| Property              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IncomingScript        | Optional script Prime Access Registrar server runs when it receives a request from a client for EAP-TLS service                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| OutgoingScript        | Optional script Prime Access Registrar server runs before it sends a response to a client using EAP-TLS                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| MaximumMessageSize    | Indicates the maximum length in bytes that a PEAP or EAP-TLS message can have before it is fragmented.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| PrivateKeyPassword    | The password used to protect the server's private key.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| ServerCertificateFile | The full pathname of the file containing the server's certificate or certificate chain used during the TLS exchange. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The two valid encoding prefixes are PEM and DER. If an encoding prefix is not present, the file is assumed to be in PEM format.                                                                                                                                                                                                                                                                                                                                                            |
| ServerKeyFile         | The full pathname of the file containing the server's RSA or ECC ( <a href="#">remove for Diameter</a> ) private key. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The two valid encoding prefixes are "PEM" and "DER". If an encoding prefix is not present, the file is assumed to be in PEM format.<br><br>The following example assumes that the subdirectory <code>pki</code> under <code>/cisco-ar</code> contains the server's certificate file. The file <code>server-key.pem</code> is assumed to be in PEM format. The file extension <code>.pem</code> is not significant.<br><br><code>set ServerKeyFile PEM:/cisco-ar/pki/server-key.pem</code> |

Table 5-7 EAP-TLS Service Properties (continued)

| Property               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CACertificateFile      | The full pathname of the file containing trusted CA certificates used for client verification. The file can contain more than one certificate, but all certificates must be in PEM format. DER encoding is not allowed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| CACertificatePath      | <p>The name of a directory containing trusted CA certificates (in PEM format) used for client verification. This parameter is optional, and if it is used there are some special preparations required for the directory it references.</p> <p>Each certificate file in this directory must contain exactly one certificate in PEM format. The server looks up the certificate files using the MD5 hash value of the certificate's subject name as a key. The directory must therefore also contain a set of symbolic links each of which points to an actual certificate file. The name of each symbolic link is the hash of the subject name of the certificate.</p> <p>For example, if a certificate file named <b>ca-cert.pem</b> is located in the CACertificatePath directory, and the MD5 hash of the subject name contained in <b>ca-cert.path.pem</b> is 1b96dd93, then a symbolic link named 1b96dd93 must point to <b>ca-cert.pem</b>.</p> <p>If there are subject name collisions such as multiple certificates with the same subject name, each link name must be indexed with a numeric extension as in 1b96dd93.0 and 1b96dd93.1.</p> |
| CRLDistributionURL     | <p>Optional. The URL that Prime Access Registrar should use to retrieve the CRL. You can specify a URL that uses HTTP or LDAP.</p> <p>The following is an example for an HTTP URL:<br/>&lt;http://crl.verisign.com/pca1.1.1.crl&gt;.</p> <p>The following is an example for an LDAP URL:<br/>ldap://209.165.200.225:388/CN=development-CA,CN=acs-westcoast2,CN=CDP,CN=Public Key Services,CN=Services,CN=Configuration,DC=cisco,DC=com</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| ClientVerificationMode | <p>Specifies the type of verification used for client certificates. Must be set to one of RequireCertificate, None, or Optional.</p> <ul style="list-style-type: none"> <li>RequireCertificate causes the server to request a client certificate and authentication fails if the client refuses to provide one.</li> <li>None will not request a client certificate.</li> <li>Optional causes the server to request a client certificate but the client is allowed to refuse to provide one.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| VerificationDepth      | Specifies the maximum length (in bytes?) of the certificate chain used for client verification.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

**Table 5-7 EAP-TLS Service Properties (continued)**

| Property              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UseECCCertificates    | <p>Determines the applicability of the authentication mechanism in SmartGrid Solutions, see the <a href="#">Smart Grid Solution Management, page 9-51</a> for more information.</p> <p>When UseECCCertificates is set to True, it can use the ECC, RSA, or combination of both certificate for certificate based verification.</p> <p>When UseECCCertificates is set to False, it can only use the RSA certificate for certificate based verification. The default location to fetch the certificate file is <b>/cisco-ar/pki</b>.</p> |
| EnableSessionCache    | Specifies whether TLS session caching (fast reconnect) is enabled or not. Set to True to enable session caching; otherwise set to False.                                                                                                                                                                                                                                                                                                                                                                                               |
| SessionTimeout        | <p>If TLS session caching (fast reconnect) is enabled, SessionTimeout specifies the maximum lifetime of a TLS session. Expired sessions are removed from the cache and will require a subsequent full authentication.</p> <p>SessionTimeout is specified as a string consisting of pairs of numbers and units, where units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks, as in the following:</p> <p style="text-align: center;"><b>Set SessionTimeout “1 Hour 45 Minutes”</b></p>  |
| AuthenticationTimeout | Mandatory; specifies time (in seconds) to wait before an authentication request times out; defaults to 120.                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Enable autochaining   | When set to TRUE, Prime Access Registrar sends its server certificate chain (Server-Cert -> IntermediateCA -> RootCA) while presenting the server certificate to the client for server side authentication. When set to FALSE, Prime Access Registrar sends only the server certificate (Server-Cert) to the client.                                                                                                                                                                                                                   |

To enable EAP-TLS authentication:

**Step 1** Launch **aregcmd** and create an EAP-TLS service.

```
cd /Radius/Services
```

```
add eap-tls-service
```

**Step 2** Change directory to the service and set its type to eap-tls.

```
cd eap-tls-service
```

```
set Type eap-tls
```

```
[ //localhost/Radius/Services/eap-tls-service ]
Name = eap-tls-service
Description =
Type = eap-tls
IncomingScript~ =
OutgoingScript~ =
MaximumMessageSize = 1024
PrivateKeyPassword = <encrypted>
ServerCertificateFile = /opt/CSC0ar/pki/server-cert.pem
```

```

ServerKeyFile = /opt/CSC0ar/pki/server-key.pem
CACertificateFile = /opt/CSC0ar/pki/root-cert.pem
CACertificatePath = /opt/CSC0ar/pki
CRLDistributionURL =
ClientVerificationMode = Optional
VerificationDepth = 4
EnableSessionCache = true
UseECCCertificates = true
SessionTimeout = "5 Minutes"
AuthenticationTimeout = 120

```

**Note**

Prime Access Registrar verifies the certificate during the TLS-based authentication. CRL validation is done before accepting a client certificate during the TLS authentication.

## Configuring EAP-TLS with OCSP Support

You can configure an EAP-TLS service to support Online Certificate Status Protocol (OCSP), which is used to check the status of X.509 digital certificates. This protocol can be used as an alternate to the certificate revocation list (CRL). For more information on CRL, see [CRL Support for Cisco Prime Access Registrar, page 5-58](#).

Prime Access Registrar queries any number of OCSP servers to check the revocation status based on the URLs present in the incoming packet.

OCSP can return the following three values for a given certificate request:

- Good—The certificate is good for usage. This OCSP response is taken as a final response and Access-Accept will be sent to the client.
- Revoked—The certificate is revoked. This OCSP response is taken as a final response and Access-Reject will be sent to the client.
- Unknown —If the certificate status is unknown or if none of the OCSP servers respond, failover to CRL happens. In that case, response from CRL is considered as final and an Access-Accept or Access-Reject is sent to the client accordingly.

[Table 5-8](#) describes the EAP-TLS configuration property with OCSP support:

**Table 5-8** EAP-TLS Service Property with OCSP Support

| Property               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ClientVerificationMode | <p>Specifies the type of verification used for client certificates. Must be set to one of the following:</p> <ul style="list-style-type: none"> <li>• RequireCertificate—Causes the server to request a client certificate and authentication fails if the client refuses to provide one.</li> <li>• None—Server will not request a client certificate.</li> <li>• Optional—Causes the server to request a client certificate but the client is allowed to refuse to provide one.</li> </ul> |

## Testing EAP-TLS with RSA or ECC Certificate using radclient

To test the EAP-TLS service, launch **radclient** and use the **simple\_eap\_tls\_test** command, as in the following:

```
simple_eap_tls_test arg1
```

The argument is arbitrary for the **simple\_eap\_tls\_test** command and can be anything. You can either select RSA or ECC client certificates using this argument.

## Testing EAP-TLS with Client Certificates

You can test EAP-TLS using client certificates verified by the server during the TLS exchange. The client certificate file and RSA or ECC key file must reside in **/cisco-ar/pki** and be named **client-cert.pem** and **client-key.pem** respectively. Both files must be in PEM format.

## EAP-TTLS

Prime Access Registrar supports the Extensible Authentication Protocol Tunneled TLS (EAP-TTLS). EAP-TTLS is an EAP protocol that extends EAP-TLS. In EAP-TLS, a TLS handshake is used to mutually authenticate a client and server. EAP-TTLS extends this authentication negotiation by using the secure connection established by the TLS handshake to exchange additional information between client and server.

EAP-TTLS leverages TLS (RFC 2246) to achieve certificate-based authentication of the server (and optionally the client) and creation of a secure session that can then be used to authentication the client using a legacy mechanism. EAP-TTLS provides several benefits:

- Industry standard authentication of the server using certificates (TLS)
- Standardized method for session key generation using TLS PRF
- Strong mutual authentication
- Identity privacy
- Fast reconnect using TLS session caching
- EAP message fragmentation
- Secure support for legacy client authentication methods

EAP-TTLS is a two-phase protocol. Phase 1 conducts a complete TLS session and derives the session keys used in Phase 2 to securely tunnel attributes between the server and the client. The attributes tunneled during Phase 2 can be used to perform additional authentication(s) via a number of different mechanisms.

The authentication mechanisms that can be used during Phase 2 include PAP, CHAP, MS-CHAP, MS-CHAPv2, and EAP. If the mechanism is EAP, then several different EAP methods are possible.

The Phase 2 authentication can be performed by the local AAA Server (the same server running EAP-TTLS) or it can be forwarded to another server (known as the home AAA Server). In the latter case, the home server has no involvement in the EAP-TTLS protocol and can be any AAA service that understands the authentication mechanism in use and is able to authenticate the user. It is not necessary for the home server to understand EAP-TTLS.

This section contains the following topics:

- [Configuring EAP-TTLS](#)
- [Testing EAP-TTLS with radclient](#)

## Configuring EAP-TTLS

Configuring EAP-TTLS involves two major tasks:

1. Configuring the TLS parameters used for Phase 1
2. Selecting the Phase 2 authentication methods and specifying whether authentication is performed locally or forwarded to the home server.

If authentication is forwarded, the configuration must include the identity of the remote home server and its shared secret.

You configure EAP-TTLS using the **aregcmd** CLI to create the appropriate services and specify their parameters. Use the **radclient** test tool to confirm that the services have been properly configured and are operational.

## Creating an EAP-TTLS Service

You can use **aregcmd** to create and configure a service of type *eap-ttls*. [Table 5-9](#) describes the EAP-TTLS configuration properties:

**Table 5-9 EAP-TTLS Service Properties**

| Property              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IncomingScript        | Optional script Prime Access Registrar server runs when it receives a request from a client for EAP-TTLS service.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| OutgoingScript        | Optional script Prime Access Registrar server runs before it sends a response to a client using EAP-TTLS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| MaximumMessageSize    | Indicates the maximum length in bytes that a PEAP or EAP-TLS message can have before it is fragmented.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| PrivateKeyPassword    | The password used to protect the server's private key.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| ServerCertificateFile | The full pathname of the file containing the server's certificate or certificate chain used during the TLS exchange. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The two valid encoding prefixes are PEM and DER. If an encoding prefix is not present, the file is assumed to be in PEM format.                                                                                                                                                                                                                                                                                      |
| ServerKeyFile         | The full pathname of the file containing the server's RSA or ECC private key. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The two valid encoding prefixes are "PEM" and "DER". If an encoding prefix is not present, the file is assumed to be in PEM format.<br><br>The following example assumes that the subdirectory <b>pki</b> under <b>/cisco-ar</b> contains the server's certificate file. The file <b>server-key.pem</b> is assumed to be in PEM format. The file extension <b>.pem</b> is not significant.<br><br><b>set ServerKeyFile PEM:/cisco-ar/pki/server-key.pem</b> |

Table 5-9 EAP-TTLS Service Properties (continued)

| Property               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CACertificateFile      | <p>The full pathname of the file containing trusted CA certificates used for client verification. The file can contain more than one certificate, but all certificates must be in PEM format.</p> <p><b>Note</b> DER encoding is not allowed.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| CACertificatePath      | <p>The name of a directory containing trusted CA certificates (in PEM format) used for client verification. This parameter is optional, and if used, there are some special preparations required for the directory it references.</p> <p>Each certificate file in this directory must contain exactly one certificate in PEM format. The server looks up the certificate files using the MD5 hash value of the certificate's subject name as a key. The directory must therefore also contain a set of symbolic links each of which points to an actual certificate file. The name of each symbolic link is the hash of the subject name of the certificate.</p> <p>For example, if a certificate file named <b>ca-cert.pem</b> is located in the CACertificatePath directory, and the MD5 hash of the subject name contained in <b>ca-cert.path.pem</b> is 1b96dd93, then a symbolic link named 1b96dd93 must point to <b>ca-cert.pem</b>.</p> <p>If there are subject name collisions such as multiple certificates with the same subject name, each link name must be indexed with a numeric extension as in 1b96dd93.0 and 1b96dd93.1.</p> <p>See <a href="#">rehash-ca-certs Utility</a>, page 5-44 for information about how to create the required certificate file hash links.</p> |
| CRLDistributionURL     | <p>Optional. The URL that Prime Access Registrar should use to retrieve the CRL. You can specify a URL that uses HTTP or LDAP.</p> <p>The following is an example for an HTTP URL:<br/>&lt;http://crl.verisign.com/pca1.1.1.crl&gt;.</p> <p>The following is an example for an LDAP URL:<br/>ldap://209.165.200.225:388/CN=development-CA,CN=acs-westcoast2,CN=CDP,CN=Public Key Services,CN=Services,CN=Configuration,DC=cisco,DC=com</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| ClientVerificationMode | <p>Specifies the type of verification used for client certificates. Must be set to one of RequireCertificate, None, or Optional.</p> <ul style="list-style-type: none"> <li>• RequireCertificate causes the server to request a client certificate and authentication fails if the client refuses to provide one.</li> <li>• None will not request a client certificate.</li> <li>• Optional causes the server to request a client certificate but the client is allowed to refuse to provide one.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| VerificationDepth      | <p>Specifies the maximum length of the certificate chain used for client verification.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

**Table 5-9 EAP-TTLS Service Properties (continued)**

| <b>Property</b>       | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UseECCCertificates    | <p>Determines the applicability of the authentication mechanism in SmartGrid Solutions, see the <a href="#">Smart Grid Solution Management, page 9-51</a> for more information.</p> <p>When UseECCCertificates is set to True, it can use the ECC, RSA, or combination of both certificate for certificate based verification.</p> <p>When UseECCCertificates is set to False, it can only use the RSA certificate for certificate based verification. The default location to fetch the certificate file is <b>/cisco-ar/pki</b>.</p> |
| EnableSessionCache    | <p>Specifies whether TLS session caching (fast reconnect) is enabled or not. Set to True to enable session caching; otherwise set to False.</p>                                                                                                                                                                                                                                                                                                                                                                                        |
| SessionTimeout        | <p>If TLS session caching (fast reconnect) is enabled, SessionTimeout specifies the maximum lifetime of a TLS session. Expired sessions are removed from the cache and require a subsequent full authentication.</p> <p>SessionTimeout is specified as a string consisting of pairs of numbers and units, where units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks, as in the following:</p> <p style="text-align: center;"><b>Set SessionTimeout “1 Hour 45 Minutes”</b></p>       |
| AuthenticationTimeout | <p>Mandatory; specifies time (in seconds) to wait before an authentication request times out. The default is 120.</p>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| AuthenticationService | <p>Mandatory; specifies the authentication service to use to authenticate users. See <a href="#">Configuring an EAP-TTLS Authentication Service, page 5-39</a> for more information.</p> <p><b>Note</b> The authentication service must exist before you can save the EAP-TTLS service configuration.</p>                                                                                                                                                                                                                              |

To enable EAP-TTLS authentication:

**Step 1** Launch **aregcmd** and create an EAP-TTLS service.

```
cd /Radius/Services
```

```
add eap-ttls-service
```

**Step 2** Change directory to the service and set its type to eap-ttls.

```
cd eap-ttls-service
```

```
set Type eap-ttls
```

```
[ //localhost/Radius/Services/eap-ttls-service ]
Name = eap-ttls-service
Description =
Type = eap-ttls
IncomingScript~ =
OutgoingScript~ =
MaximumMessageSize = 1024
PrivateKeyPassword = <encrypted>
ServerCertificateFile = /opt/CSC0ar/pki/server-cert.pem
ServerKeyFile = /opt/CSC0ar/pki/server-key.pem
CACertificateFile = /opt/CSC0ar/pki/root-cert.pem
CACertificatePath = /opt/CSC0ar/pki
CRLDistributionURL =
ClientVerificationMode = Optional
VerificationDepth = 4
EnableSessionCache = true
UseECCCertificates = true
SessionTimeout = "5 Minutes"
AuthenticationTimeout = 120
```



**Note**

Prime Access Registrar verifies the certificate during the TLS-based authentication. CRL validation is done before accepting a client certificate during the TLS authentication.

## Configuring an EAP-TTLS Authentication Service

The EAP-TTLS service can authenticate users with either a legacy method such as PAP, CHAP, MSCHAP, or MSCHAPv2 or with an EAP method such as EAP-MSCHAPv2 or EAP-GTC. The authentication can be performed by the local server (the same server running EAP-TTLS) or it can be forwarded to a remote AAA Server (the home server for the user's domain).

This section provides examples of several different ways to configure an EAP-TTLS authentication service. The following examples assume that you are using aregcmd and have already created the EAP-TTLS service.



**Note**

After you make a configuration change, you must save the configuration before it can be used.

### Authenticating Local Users with a Legacy Method

You can use a service like the local-users service (created as part of the example configuration) to authenticate users in the local UserList.

```
set AuthenticationService local-users
```

This service can be used to authenticate using PAP, CHAP, MSCHAP, and MSCHAPv2.

### Authenticating Users with EAP-MSChapV2

This example uses a service named eap-mschapv2 for authentication. Attempts to authenticate using any other method than EAP-MSChapV2 (assuming the service type is also eap-mschapv2) will fail.

```
set AuthenticationService eap-mschapv2
```

### Authenticating Users with EAP Negotiate

You can use the EAP-negotiate method to authenticate using more than one EAP type. The following example defines an EAP service named eap-negotiate that can negotiate EAP-MSChapV2 or EAP-GTC then configures an EAP-TTLS service to authenticate using that service.

To configure an EAP-TTLS service to authenticate using eap-negotiate:

- 
- Step 1** Create a service of type *eap-negotiate*.
- ```
cd /Radius/Services
add eap-nego
cd eap-nego
set Type eap-negotiate
set ServiceList "eap-mschapv2 eap-gtc"
```
- Step 2** Configure the EAP-TTLS AuthenticationService.
- ```
cd /Radius/Services/eap-ttls
set AuthenticationService eap-nego
```
- 

### Authenticating Users with Legacy and EAP Methods

You can configure EAP-TTLS to authenticate using both legacy and EAP methods with a Group service using an OR result rule. A configuration like that shown in the following example first attempts to authenticate with the eap-negotiate service. If that fails, the server attempts to authenticate with the local-users service.

To authenticate with the eap-negotiate service;

- 
- Step 1** Create the Group service
- ```
cd /Radius/Services
add local-or-eap
cd local-or-eap
set Type group
set ResultRule OR
cd GroupServices
add 1 eap-negotiate
add 2 local-users
```
- Step 2** Configure the EAP-TTLS AuthenticationService.
- ```
cd /Radius/Services/eap-ttls
set AuthenticationService local-or-eap
```
- 

### Authenticating Using a Remote AAA Server

You can configure an EAP-TTLS service to forward authentication to a remote AAA Server known (or the home server). The following configures a RADIUS service to use a remote server, then configures EAP-TTLS to use that service for authentication.

The first step in the following example configures a remote RADIUS server (aaa-remote) with its IP address and the shared secret that it shares with the local server. You might also specify other important parameters such as ports, timeouts, and maximum number of retries. See , for information about configuring RADIUS services.

To configure a remote RADIUS server (aaa-remote) with its IP address and a shared secret:

- 
- Step 1** Configure a remote AAA Server.
- ```
cd /Radius/RemoteServers
add aaa-remote
cd aaa-remote
set Protocol Radius
set IPAddress 10.1.2.3
set SharedSecret secret
```

The following step configures a RADIUS service to use the remote server created in the previous step. You might also configure other important parameters such as the failover strategy. See , for information about configuring RADIUS services.

**Step 2** Configure an AAA service.

```
cd /Radius/Services
add home
cd home
set Type Radius
cd RemoteServers
add 1 aaa-remote
```

**Step 3** Configure the EAP-TTLS AuthenticationService:

```
cd /Radius/Services/eap-ttls
set AuthenticationService home
```

---

Other configurations are also possible. For example, a group service can be used to perform some authentications locally and forward others to a remote server.

## Testing EAP-TTLS with radclient

To test the EAP-TLS service, launch **radclient** and use the **simple\_eap\_ttls\_test** command. The **simple\_eap\_ttls\_test** command has the following syntax:

```
simple_eap_ttls_test identity password { method }
```

Where:

*identity* is the user's name.

*password* is the user's password

*method* is one of: PAP, CHAP, MSChap, MSChapV2, or PEAP.




---

**Note** If the method parameter is EAP, the **tunnel** command must be used to specify the EAP method type.

---

## Testing EAP-TTLS Using Legacy Methods

To authenticate a user using EAP-TTLS with PAP:

---

**Step 1** Launch **radclient**.

```
cd /cisco-ar/usrbin
./radclient -s
```

**Step 2** Authenticate using EAP-TTLS PAP.

```
simple_eap_tls_test bob bob pap
```

The following commands show how to test the other valid legacy methods.

```
simple_eap_tls_test bob bob chap
simple_eap_tls_test bob bob mschap
simple_eap_tls_test bob bob mschapv2
```

---

## Testing EAP-TTLS Using EAP Methods

The following example uses EAP-TTLS with EAP-MSChapV2 as the Phase 2 method to authenticate a user named bob whose password is bob (from the example configuration). Issue the **tunnel** command to specify the Phase 2 EAP method, then issue the **simple\_eap\_tls\_test** command with eap as a method type.

To authenticate a user using EAP-TTLS with EAP-MSChapV2 as the Phase 2 method:

---

**Step 1** Launch **radclient**

```
cd /cisco-ar/usrbin
./radclient -s
```

**Step 2** Authenticate using EAP-TTLS and EAP-MSChapV2.

```
tunnel eap-mschapv2
simple_eap_tls_test bob bob eap
```

To test with a different EAP method, use the **tunnel** command to specify the method as shown in the following command to specify EAP-TLS.

```
tunnel eap-tls
simple_eap_tls_test bob bob eap
```

---

## rehash-ca-certs Utility

The **rehash-ca-certs** utility works with the `CACertificatePath` property and enables you to create the required certificate file hash links (similar to those used with PEAP and EAP-TLS). The **rehash-ca-certs** utility is only used when the server is validating certificates from the client (which is optional and not a common case for EAP-TTLS).

The syntax for the **rehash-ca-certs** utility is:

```
rehash-ca-certs { -v } path1 { path2 ... pathn }
```

Each directory path specified on the command line is scanned by the **rehash-ca-certs** utility for filenames with the **pem** extension (such as **ca-cert.pem**) and the appropriate hash link is created as described above. Before creating links, **rehash-ca-certs** first removes all existing links in the directory, so each invocation creates fresh links. The `-v` option enables verbose output.

The following is an example of the **rehash-ca-certs** utility:

```
./rehash-ca-certs ../pki  
  
start rehashing ../pki  
client-key.pem does not contain a PEM certificate  
finished rehashing
```

The **rehash-ca-certs** utility warns about PEM files that do not contain certificates. On Cisco Prime Access Registrar, intermediate/chained certificates cannot be imported.

To run Prime Access Registrar with PEAP authentication:

- 
- Step 1** Add both root and intermediate CA in the directory `/opt/CSCOAr/pki` (as configured for `CACertificatePath` in the service `NYU-NetIDs-PEAPService`).
  - Step 2** Change the directory to `pki`:

```
cd /opt/CSCOAr/pki
```
  - Step 3** run `/opt/CSCOAr/bin/rehash-ca-certs`
  - Step 4** Stop ARserver and restart.
- 

## radclient Command Reference

This section provides a summary of the **radclient** commands you can use to test PEAP and EAP-TLS. It contains the following topics:

- [eap-trace](#)
- [tunnel](#)

## eap-trace

Use the **eap-trace** command to display additional client protocol trace information for EAP methods. Set the level to a number from 1 to 5 inclusively. Level 5 shows detailed hexadecimal dumps of all messages. Level 4 shows a message trace without hexadecimal dumps. Levels 3 and below show status and error information. To turn off trace displays, set the level to 0.

Use **eap-trace level** to set the trace level for all EAP methods. The following example command sets the trace level to 4 for all EAP methods:

```
eap-trace 4
```

Use **eap-trace method level** to set the trace level for the specified EAP method. The following example command sets the trace level to 5 for PEAP Version0 only. The trace level for other EAP methods is not affected.

```
eap-trace peap-v0 5
```

**Note**

---

The **eap-trace** command is for client-side trace information only and is independent of the server trace level you set using **aregcmd**.

---

## tunnel

Use the **tunnel** command to specify the inner authentication method for PEAP. The specified EAP method type must agree with the server's configured authentication method or authentication will fail.

```
tunnel eap-method
```

For PEAP Version 0, the allowable tunnel methods are EAP-MSCHAPV2 and EAP-SIM. For PEAP Version 1, the allowable tunnel methods are EAP-GTC and EAP-SIM.

```
simple_eap_mschapv2_test username password
```

```
simple_eap_gtc_test username password
```

```
simple_eap_peapv0_test arg1 arg2
```

The arguments are passed to the inner authentication method as its authentication parameters. For EAP-MSChapv2 the arguments are username and password; for EAP-SIM they are IMSI and key.

```
simple_eap_peapv1_test arg1 arg2
```

The arguments are passed to the inner authentication method as its authentication parameters. For EAP-GTC the arguments are username and password; for EAP-SIM they are IMSI and key.

```
simple_eap_tls_test arg1
```

## Protected EAP

Protected EAP (PEAP) is an authentication method designed to mitigate several weaknesses of EAP. PEAP leverages TLS (RFC 2246) to achieve certificate-based authentication of the server (and optionally the client) and creation of a secure session that can then be used to authenticate the client. PEAP provides several benefits:

- Industry standard authentication of the server using certificates (TLS)
- Standardized method for session key generation using TLS PRF
- Strong mutual authentication
- Identity privacy
- Fast reconnect using TLS session caching
- EAP message fragmentation
- Secure support for legacy client authentication methods

Cisco Prime Access Registrar supports the two major existing variants of PEAP, PEAP Version 0 (Microsoft PEAP) and PEAP Version 1 (Cisco PEAP). PEAP Version 0 is described in IETF drafts, **draft-kamath-pppext-peapv0-00.txt** and **draft-josefsson-pppext-eap-tls-eap-02.txt**. This version of PEAP can use either EAP-MSChapV2 or EAP-SIM as an authentication method. PEAP Version 1 is described by IETF draft **draft-zhou-pppext-peapv1-00.txt**. PEAP Version 1 can use either EAP-GTC or EAP-SIM as an authentication method.

This section contains the following topics:

- [PEAP Version 0](#)
- [PEAP Version 1](#)

## PEAP Version 0

This section describes configuring PEAP Version 0 and testing it with **radclient**.

### Configuring PEAP Version 0

You can use **aregcmd** to create and configure a service of type *peap-v0*. [Table 5-10](#) describes the PEAP service properties for PEAP Version 0.

**Table 5-10** PEAP Version 0 Service Properties

Property	Description
IncomingScript	Optional script Prime Access Registrar server runs when it receives a request from a client for PEAP-v0 service.
OutgoingScript	Optional script Prime Access Registrar server runs before it sends a response to a client using PEAP-v0
MaximumMessageSize	Indicates the maximum length in bytes that a PEAP or EAP-TLS message can have before it is fragmented.
PrivateKeyPassword	The password used to protect the server's private key.

Table 5-10 PEAP Version 0 Service Properties (continued)

Property	Description
ServerCertificateFile	<p>The full pathname of the file containing the server's certificate or certificate chain used during the TLS exchange. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The two valid encoding prefixes are PEM and DER. If an encoding prefix is not present, the file is assumed to be in PEM format.</p> <p>The following example assumes that the subdirectory <b>pki</b> under <b>/cisco-ar</b> contains the server's certificate file. The file <b>server-cert.pem</b> is assumed to be in PEM format; note that the file extension <b>.pem</b> is not significant.</p> <pre>set ServerCertificateFile PEM:/cisco-ar/pki/server-cert.pem</pre>
CACertificateFile	<p>The full pathname of the file containing trusted CA certificates used for client verification. The file can contain more than one certificate, but all certificates must be in PEM format. DER encoding is not allowed.</p>
CACertificatePath	<p>The name of a directory containing trusted CA certificates (in PEM format) used for client verification. This parameter is optional, and if it is used there are some special preparations required for the directory it references.</p> <p>Each certificate file in this directory must contain exactly one certificate in PEM format. The server looks up the certificate files using the MD5 hash value of the certificate's subject name as a key. The directory must therefore also contain a set of symbolic links each of which points to an actual certificate file. The name of each symbolic link is the hash of the subject name of the certificate.</p> <p>For example, if a certificate file name <b>ca-cert.pem</b> is located in the CACertificatePath directory, and the MD5 hash of the subject name contained in <b>ca-cert.path.pem</b> is 1b96dd93, then a symbolic link named 1b96dd93 must point to the <b>ca-cert.pem</b> file.</p> <p>If there are subject name collisions such as multiple certificates with the same subject name, each link name must be indexed with a numeric extension as in 1b96dd93.0 and 1b96dd93.1.</p>
CRLDistributionURL	<p>Optional. The URL that Prime Access Registrar should use to retrieve the CRL. You can specify a URL that uses HTTP or LDAP.</p> <p>The following is an example for an HTTP URL: &lt;http://crl.verisign.com/pca1.1.1.crl&gt;.</p> <p>The following is an example for an LDAP URL: ldap://209.165.200.225:388/CN=development-CA,CN=acs-westcoast2,CN=CDP,CN=Public Key Services,CN=Services,CN=Configuration,DC=cisco,DC=com</p>

Table 5-10 PEAP Version 0 Service Properties (continued)

Property	Description
ClientVerificationMode	<p>Specifies the type of verification used for client certificates. Must be set to one of RequireCertificate, None, or Optional.</p> <ul style="list-style-type: none"> <li>RequireCertificate causes the server to request a client certificate and authentication fails if the client refuses to provide one.</li> <li>None will not request a client certificate.</li> <li>Optional causes the server to request a client certificate but the client is allowed to refuse to provide one.</li> </ul>
VerificationDepth	<p>Specifies the maximum length of the certificate chain used for client verification.</p>
UseECCCertificates	<p>Determines the applicability of the authentication mechanism in SmartGrid Solutions, see the <a href="#">Smart Grid Solution Management, page 9-51</a> for more information.</p> <p>When UseECCCertificates is set to True, it can use the ECC, RSA, or combination of both certificate for certificate based verification.</p> <p>When UseECCCertificates is set to False, it can only use the RSA certificate for certificate based verification. The default location to fetch the certificate file is <b>/cisco-ar/pki</b>.</p>
EnableSessionCache	<p>Specifies whether TLS session caching (fast reconnect) is enabled or not. Set to True to enable session caching; otherwise set to False.</p>
SessionTimeout	<p>If TLS session caching (fast reconnect) is enabled, SessionTimeout specifies the maximum lifetime of a TLS session. Expired sessions are removed from the cache and will require a subsequent full authentication.</p> <p>SessionTimeout is specified as a string consisting of pairs of numbers and units, where units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks, as in the following:</p> <p style="text-align: center;"><b>Set SessionTimeout “1 Hour 45 Minutes”</b></p>
AuthenticationTimeout	<p>Mandatory; specifies time (in seconds) to wait before an authentication request times out; defaults to 120.</p>
TunnelService	<p>Mandatory; must be the name of an existing EAP-MSCHAPv2 or EAP-SIM service for PEAP Version 0.</p>
EnableWPS	<p>When set to TRUE, enables Windows Provisioning Service (WPS) and provides two other properties, MasterURL and WPSGuestUserProfile. The default value is FALSE.</p>

**Table 5-10 PEAP Version 0 Service Properties (continued)**

Property	Description
MasterURL	When using WPS, specifies the URL of the provisioning server which is modified with the appropriate fragment and sent to the client.
WPSGuestUserProfile	When using WPS, specifies a profile to be used as a guest user profile; must be a valid profile under <b>/Radius/Profiles</b> .  This profile is used for guests and users whose account has expired. This profile normally contains attributes denoting the VLAN-id of the guest network (which has the provisioning server alone) and might contain IP-Filters that would restrict the access of the guest (to only the provisioning server).

To enable PEAP Version 0:

**Step 1** Launch **aregcmd** and create a PEAP Version 0 service.

```
cd /Radius/Services
add peap-v0-service
```

**Step 2** Set the service's type to peap-v0.

```
cd peap-v0-service
set Type peap-v0
```

```
//localhost/Radius/Services/peap-v0-service ]
Name = peap-v0-service
Description =
Type = peap-v0
IncomingScript~ =
OutgoingScript~ =
MaximumMessageSize = 1024
PrivateKeyPassword = <encrypted>
ServerCertificateFile = /opt/CSCOar/pki/server-cert.pem
ServerKeyFile = /opt/CSCOar/pki/server-key.pem
CACertificateFile = /opt/CSCOar/pki/root-cert.pem
CACertificatePath = /opt/CSCOar/pki
CRLDistributionURL =
ClientVerificationMode = Optional
VerificationDepth = 4
EnableSessionCache = true
UseECCCertificates = true
SessionTimeout = "5 Minutes"
AuthenticationTimeout = 120
EnableWPS = FALSE
```

**Step 3** Set the service's TunnelService property to the name of an existing EAP-MSCHAPV2 or EAP-SIM service.

```
set TunnelService name_of_EAP-MSCHAPv2_service

or
```

```
set TunnelService name_of_EAP-SIM_service
```

---

**Note**

Prime Access Registrar verifies the certificate during the TLS-based authentication. CRL validation is done before accepting a client certificate during the TLS authentication.

---

## Testing PEAP Version 0 with radclient

To test the PEAP Version 0:

---

**Step 1** Launch **radclient**.

**Step 2** Specify the inner authentication method, `eap-mschapv2` or `eap-sim`, as in the following.

```
tunnel eap-mschapv2
```

*or*

```
tunnel eap-sim
```

**Step 3** Use the `simple_eap_peapv0_test` command to authenticate using PEAP Version 0, as in the following:

```
simple_eap_peapv0_test arg1 arg2
```

The `simple_eap_peapv0_test` command passes its arguments to the inner authentication mechanism which treats the arguments as either a username and a password (for `eap-mschapv2`) or as an IMSI and a key (for `eap-sim`).

---

The following example tests PEAP Version 0 with EAP-MSCHAPV2 as the inner authentication mechanism using username bob and password bob:

```
tunnel eap-mschapv2
```

```
simple_eap_peapv0_test bob bob
```

The following example tests PEAP Version 0 with EAP-SIM as the inner authentication mechanism using IMSI 1124567891 and key 0112456789ABCDEF:

```
tunnel eap-sim
```

```
simple_eap_peapv0_test 1124567891 0112456789ABCDEF
```

---

## Testing PEAP Version 0 with Client Certificates

You can test PEAP Version 0 using client certificates verified by the server during the TLS exchange. The client certificate file and RSA or ECC key file must reside in `/cisco-ar/pki` and be named `client-cert.pem` and `client-key.pem` respectively. Both files must be in PEM format.

## PEAP Version 1

This section describes configuring PEAP Version 1 and testing it with **radclient**.

### Configuring PEAP Version 1

You can use **aregcmd** to create and configure a service of type *peap-v1*. Table 5-11 describes the PEAP service properties for both PEAP Version 1.

**Table 5-11 PEAP Version 1 Service Properties**

Property	Description
IncomingScript	Optional script Prime Access Registrar server runs when it receives a request from a client for PEAP-v1 service.
OutgoingScript	Optional script Prime Access Registrar server runs before it sends a response to a client using PEAP-v1.
MaximumMessageSize	Indicates the maximum length in bytes that a PEAP or EAP-TLS message can have before it is fragmented.
PrivateKeyPassword	The password used to protect the server's private key.
ServerCertificateFile	The full pathname of the file containing the server's certificate or certificate chain used during the TLS exchange. The pathname can be optionally prefixed with a special string that indicates the type of encoding used for the certificate. The two valid encoding prefixes are PEM and DER. If an encoding prefix is not present, the file is assumed to be in PEM format.
CACertificateFile	The full pathname of the file containing trusted CA certificates used for client verification. The file can contain more than one certificate but all certificates must be in PEM format. DER encoding is not allowed.
CACertificatePath	<p>The name of a directory containing trusted CA certificates (in PEM format) used for client verification. This parameter is optional, and if it is used there are some special preparations required for the directory it references.</p> <p>Each certificate file in this directory must contain exactly one certificate in PEM format. The server looks up the certificate files using the MD5 hash value of the certificate's subject name as a key. The directory must therefore also contain a set of symbolic links each of which points to an actual certificate file. The name of each symbolic link is the hash of the subject name of the certificate.</p> <p>For example, if a certificate file named <b>ca-cert.pem</b> is located in the CACertificatePath directory, and the MD5 hash of the subject name contained in <b>ca-cert.path.pem</b> is 1b96dd93, then a symbolic link named 1b96dd93 must point to the <b>ca-cert.pem</b> file.</p> <p>If there are subject name collisions such as multiple certificates with the same subject name, each link name must be indexed with a numeric extension as in 1b96dd93.0 and 1b96dd93.1.</p>

Table 5-11 PEAP Version 1 Service Properties (continued)

Property	Description
CRLDistributionURL	<p>Optional. The URL that Prime Access Registrar should use to retrieve the CRL. You can specify a URL that uses HTTP or LDAP.</p> <p>The following is an example for an HTTP URL: &lt;http://crl.verisign.com/pca1.1.1.crl&gt;.</p> <p>The following is an example for an LDAP URL: ldap://209.165.200.225:388/CN=development-CA,CN=acs-westcoast2,CN=CDP,CN=Public Key Services,CN=Services,CN=Configuration,DC=cisco,DC=com</p>
ClientVerificationMode	<p>Specifies the type of verification used for client certificates. Must be set to one of RequireCertificate, None, or Optional.</p> <ul style="list-style-type: none"> <li>RequireCertificate causes the server to request a client certificate and authentication fails if the client refuses to provide one.</li> <li>None will not request a client certificate.</li> <li>Optional causes the server to request a client certificate but the client is allowed to refuse to provide one.</li> </ul>
VerificationDepth	Specifies the maximum length of the certificate chain used for client verification.
UseECCCertificates	<p>Determines the applicability of the authentication mechanism in SmartGrid Solutions, see the <a href="#">Smart Grid Solution Management, page 9-51</a> for more information.</p> <p>When UseECCCertificates is set to True, it can use the ECC, RSA, or combination of both certificate for certificate based verification.</p> <p>When UseECCCertificates is set to False, it can only use the RSA certificate for certificate based verification. The default location to fetch the certificate file is <b>/cisco-ar/pki</b>.</p>
EnableSessionCache	Specifies whether TLS session caching (fast reconnect) is enabled or not. Set to True to enable session caching; otherwise set to False.
SessionTimeout	<p>If TLS session caching (fast reconnect) is enabled, SessionTimeout specifies the maximum lifetime of a TLS session. Expired sessions are removed from the cache and will require a subsequent full authentication.</p> <p>SessionTimeout is specified as a string consisting of pairs of numbers and units, where units might be one of the following: M, Minute, Minutes, H, Hour, Hours, D, Day, Days, W, Week, Weeks, as in the following:</p> <p style="text-align: center;"><b>Set SessionTimeout “1 Hour 45 Minutes”</b></p>
AuthenticationTimeout	Mandatory; specifies time (in seconds) to wait before an authentication request times out; defaults to 120.
TunnelService	Mandatory; must be the name of an existing EAP-GTC or EAP-SIM service for PEAP Version 0.

To enable PEAP Version 1:

**Step 1** Launch **aregcmd** and create a PEAP Version 1 service.

```
cd /Radius/Services
add peap-v1-service
```

**Step 2** Set the service's type to peap-v1.

```
cd peap-v1-service
set Type peap-v1
```

```
//localhost/Radius/Services/eap-peap-v1-service ]
Name = eap-peap-v1-service
Description =
Type = peap-v1
IncomingScript~ =
OutgoingScript~ =
MaximumMessageSize = 1024
PrivateKeyPassword = <encrypted>
ServerCertificateFile = /opt/CSCOar/pki/server-cert.pem
ServerKeyFile = /opt/CSCOar/pki/server-key.pem
CACertificateFile = /opt/CSCOar/pki/root-cert.pem
CACertificatePath = /opt/CSCOar/pki
CRLDistributionURL =
ClientVerificationMode = Optional
VerificationDepth = 4
EnableSessionCache = true
UseECCCertificates = true
SessionTimeout = "5 Minutes"
AuthenticationTimeout = 120
```

**Step 3** Set the service's TunnelService property to the name of an existing EAP-GTC or EAP-SIM service.

```
set TunnelService name_of_EAP-GTC_service
or
set TunnelService name_of_EAP-SIM_service
```

## Testing PEAP Version 1 with radclient

To test the PEAP Version 1:

**Step 1** Launch **radclient**.

**Step 2** Specify the inner authentication method, EAP-GTC or EAP-SIM, as in the following.

```
tunnel eap-gtc
or
tunnel eap-sim
```

**Step 3** Use the `simple_eap_peapv1_test` command to authenticate using PEAP Version 1, as in the following:

```
simple_eap_peapv1_test arg1 arg2
```

The `simple_eap_peapv1_test` command passes its arguments to the inner authentication mechanism which treats the arguments as either a username and a password (for EAP-GTC) or as an IMSI and a key (for EAP-SIM).

---

## Testing PEAP Version 1 with Client Certificates

You can test PEAP Version 1 using client certificates verified by the server during the TLS exchange. The client certificate file and RSA or ECC key file must reside in `/cisco-ar/pki` and be named `client-cert.pem` and `client-key.pem` respectively. Both files must be in PEM format.

# How to Configure Oracle, Mysql Accounting with the Buffering Option Enabled

Prime Access Registrar provides support for MySQL to query user records from Oracle database using sql interface and enables you to write accounting records into Oracle database. You can use insert, update, and delete queries to

- add new details into database.
- modify the existing details in the database.
- remove the outdated details from the database.

## To Select the SQL Statement in Run Time Accounting

Prime Access Registrar provides support to query user account details from SQL database and enables you to add, delete, and update accounting details into SQL when using Oracle accounting.

You can execute the following SQL statements to perform various actions:

- [Query](#)
- [Insert](#)
- [Update](#)
- [Delete](#)
- [Configuring Oracle, Mysql Accounting](#)

## Query

You can query the accounting details from Oracle by referring this service in `/Radius/DefaultAuthenticationService` and in `/Radius/DefaultAuthorization`.

The following example is an SQL statement used for Authentication and Authorization of the subscribed users. You can use the SQL and MarkerList properties statement to query the selected attributes from Oracle.

```
sql1/  
  Name = sql1  
  Description =  
  Type = query  
  SQL = "select password , username from arusers where username = ?"  
  ExecutionSequenceNumber = 1  
  MarkerList = UserName/SQL_CHAR
```

## Insert

You can insert user details into SQL database by Oracle accounting. This service is used by referring the **/Radius/DefaultAccountingService** or **Accounting-Service** environment variable.

For instance, you can use the following SQL and MarkerList properties statement to insert the selected attributes:

```
sql1/  
  Name = sql1  
  Description =  
  Type = insert  
  SQL = "insert into sql_test (username,nas) values (?,?)"  
  ExecutionSequenceNumber = 1  
  MarkerList = "UserName/SQL_CHAR NAS-Identifier/SQL_CHAR"
```

## Update

You can easily modify the details in an SQL table with the UPDATE statement.

For example, you can use the following SQL and MarkerList properties statement to update the selected attributes:

```
sql2/  
  Name = sql2  
  Description =  
  Type = update  
  SQL = "update sql_test set packet='stop' where username=?"  
  ExecutionSequenceNumber = 2  
  MarkerList = UserName/SQL_CHAR
```

## Delete

You can remove the unnecessary records from SQL database using DELETE statement.

For example, you can use the following SQL and MarkerList properties statement to delete the selected attributes:

```
sql/  
  Name = sql  
  Description =  
  Type =delete  
  SQL = "delete from arusers_acct where username=?"  
  ExecutionSequenceNumber = 1  
  MarkerList = UserName/SQL_CHAR
```

## Configuring Oracle, Mysql Accounting

The following script describes you how to configure Oracle, Mysql accounting with the buffering option enabled:

```
[ //localhost/Radius/Services/oracle-acc ]
  Name = oracle-acc
  Description =
  Type = oci-accounting
  IncomingScript~ = sql
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
  OutageScript~ =
  MultipleServersPolicy = Failover
  RemoteServers/

[ //localhost/Radius/Services/oracle-acc/RemoteServers ]
  1. oracle-acc

[ //localhost/Radius/RemoteServers/oracle-acc ]
  Name = oracle-acc
  Description =
  Protocol = oci-accounting
  ReactivateTimerInterval = 300000
  Timeout = 15
  DataSourceConnections = 8
  ODBCDataSource = oracle
  SNMPTrapIP =
  SNMPTrapPort = 1521
  KeepAliveTimerInterval = 0
  BufferAccountingPackets = TRUE
  MaximumBufferFileSize = "10 Megabytes"
  NumberOfRetriesForBufferedPacket = 3
  BackingStoreEnvironmentVariables =
  UseLocalTimeZone = FALSE
  AttributeList =
  Delimiter =
  SQLDefinition/

[ //localhost/Radius/Advanced/ODBCDataSources/oracle ]
  Name = oracle
  Description =
  Type = oracle_oci
  UserID = scott
  Password = <encrypted>
  DataBase = ORCL

[ //localhost/Radius/Scripts/sql ]
  Name = sql
  Description =
  Language = tcl
  Filename = sql.tcl
  EntryPoint = sqltest
  InitEntryPoint =
  InitEntryPointArgs =
```

### Script

The script statements are executed based on the IP address that you specified in the query. Here is a sample script to select the SQL statements.

```
proc sqltest {request response environ} {
```

```

set nas [ $request get NAS-Identifier ]
if { [ string compare $nas 1.1.1.1 ] == 0 } {
    $enviro put SQL-Sequence "sql1"
    $enviro put BackingStore-Env-Vars "SQL-Sequence"
}
if { [ string compare $nas 1.1.1.2 ] == 0 } {
    $enviro put SQL-Sequence "sql2"
    $enviro put BackingStore-Env-Vars "SQL-Sequence"
}
if { [ string compare $nas 1.1.1.3 ] == 0 } {
    $enviro put SQL-Sequence "sql3"
    $enviro put BackingStore-Env-Vars "SQL-Sequence"
}
if { [ string compare $nas 1.1.1.4 ] == 0 } {
    $enviro put SQL-Sequence "sql4"
    $enviro put BackingStore-Env-Vars "SQL-Sequence"
}
}
}

```

## How Suffix and Prefix Rules Work with Prime Access Registrar

Prime Access Registrar includes several scripts that you can use with the rules. The following are the most commonly used rules:

- Prefix Rule, See [ExecPrefixRule, page 10-17](#) for more information
- Suffix Rule, See [ExecSuffixRule, page 10-18](#) for more information

### Configuring Prefix and Suffix Policies

To configure prefix and suffix policies in Prime Access Registrar in order to provide authentication and authorization services for the subscribed users:

- Step 1** Activate the Policy Engine by configuring **SelectPolicy**. This script explains you how to set a suffix and prefix policy in the grouping list.

```

--> cd selectPolicy/

[ //localhost/Radius/Policies/SelectPolicy ]
Name = SelectPolicy
Description =
Grouping = suffixrule&prefixrule

```

- Step 2** Run the configuration rules for Prefix and Suffix.

- Step 3** Set Script = ExecSuffixRule in the prefix rule configuration.

```

[ //localhost/Radius/Rules ]
Entries 1 to 2 from 2 total entries
Current filter: <all>

prefixrule/
Name = prefixrule
Description =
Type = radius
Script~ = ExecPrefixRule
Attributes/
Authentication-Service = local-users
Authorization-Service = local-users

```

```

Delimiters = @#%$/
Prefix = cisco
StripPrefix = no

```

**Step 4** Specify Script = ExecRealmRule in the suffix configuration to scan.

```

suffixrule/
  Name = suffixrule
  Description =
  Type = radius
  Script~ = ExecRealmRule
  Attributes/
    Realm = @cisco.com

```

## CRL Support for Cisco Prime Access Registrar

Prime Access Registrar checks for various certificates for validation purposes in its authentication services. The client sends a certificate along with the access-challenge to Prime Access Registrar. Prime Access Registrar verifies the validity of the certificate and approves the request if the certificate is valid. For certificate validation, Prime Access Registrar uses an advanced verification mechanism, which uses Certificate Revocation Lists (CRLs).

A CRL, which uses the X.509 certification format, is the signed data structure that the certificate authority (CA) issues periodically. It contains a list of the serial numbers and the timestamp of the revoked certificates. These revoked certificates are not valid and Prime Access Registrar rejects any request that comes with these certificates. The CRLs are available in a public repository in Prime Access Registrar.

A certificate can be revoked because of the following reasons:

- Expiration of the validity period.
- Change in the name of the user to whom the certificate is issued.
- Change in the association between the CA and the user.
- Loss of the private key that is associated with the certificate.

Prime Access Registrar uses the Lightweight Dynamic Authentication Protocol (LDAP) and HTTP for validating the certificates using CRL. The **CRLDistributionURL** in the TLS based EAP authentication services, is used for the CRL support in Prime Access Registrar. When you configure this property, Prime Access Registrar fetches the CRL from the specified URL, at the startup. A background thread in Prime Access Registrar keeps track of these CRLs. When any of the CRLs expires, Prime Access Registrar fetches the latest version of CRL using the specified URL. Each CRL contains the information related to its expiry.

Prime Access Registrar places all the CRLs in a CRL store. It uses these CRLs while it does a TLS authentication for certificate validation. During an authentication service, the certificate verifier in Prime Access Registrar checks for the validity of the certificate against the CRL issued by the CA that signed the certificate. It looks for the serial number of the certificate in the list of revoked certificates in the appropriate CRL. If it finds a match in the CRL, it compares the revocation time that is encoded in the CRL against the current time. If the current time is later than the revocation time, Prime Access Registrar considers the certificate invalid.

This section contains the following topics:

- [Configuring Certificate Validation Using CRL](#)
- [Using Intermediate Certificates in Prime Access Registrar](#)

**Note**

Prime Access Registrar uses the **CRLDistributionURL** property in the following services:

**eap-tls**  
**eap-ttls**  
**eap-fast**  
**peap-v0**  
**peap-v1**

## Configuring Certificate Validation Using CRL

Prime Access Registrar uses the **CRLDistributionURL** property for the certificate validation using CRLs. The following shows a sample configuration for the certificate verification using CRLs in Prime Access Registrar:

```
//localhost/Radius/Services/eap-ttls-service ]
Name = eap-ttls-service
Description =
Type = eap-ttls
IncomingScript~ =
OutgoingScript~ =
MaximumMessageSize = 1024
PrivateKeyPassword = <encrypted>
ServerCertificateFile = /opt/CSCOar/pki/server-cert.pem
ServerKeyFile = /opt/CSCOar/pki/server-key.pem
CACertificateFile = /opt/CSCOar/pki/root-cert.pem
CACertificatePath = /opt/CSCOar/pki
CRLDistributionURL =
ClientVerificationMode = Optional
VerificationDepth = 4
EnableSessionCache = true
UseECCCertificates = true
SessionTimeout = "5 Minutes"
AuthenticationTimeout = 120
```

Table 5-9 describes the properties in this sample configuration.

## Using Intermediate Certificates in Prime Access Registrar

The `rehash-ca-certs` utility can be used to import intermediate certificates in Prime Access Registrar. See [rehash-ca-certs Utility, page 5-44](#) for information about how to create the required certificate file hash links.

To import intermediate certificates in Prime Access Registrar:

- 
- Step 1** Copy the Root CA, Intermediate CA of the client to a directory.
  - Step 2** Run `/opt/CSCOar/bin/rehash-ca-certs -v <path of the client certificate store>`  
 The utility creates the required hash links to maintain the chain between the Root CA certificate and Intermediate CA certificates.
  - Step 3** Set the `CACertificateFile` property in EAP service to the path where Root CA Certificate of the client is stored.

**Step 4** Restart the Prime Access Registrar server.

---

The following shows an example to import intermediate certificates in Prime Access Registrar:

---

**Step 1** Copy the Client Root CA and Intermediate CA Certificate in `/cisco-ar/certs/wimax/` directory.

```
cp /tmp/wimax_device_root.pem /cisco-ar/certs/wimax/
cp /tmp/wimax_device_root_ca1.pem /cisco-ar/certs/wimax/
/opt/CSCOar/bin/rehash-ca-certs -v /cisco-ar/certs/wimax/
```

**Step 2** Enter in to aregcmd.

```
/opt/CSCOar/bin/aregcmd -s
```

a. Configure the eap service which uses these client certificates.

```
cd Radius/Services/eap-ttls
```

```
//localhost/Radius/Services/eap-ttls-service ]
Name = eap-ttls-service
Description =
Type = eap-ttls
IncomingScript~ =
OutgoingScript~ =
MaximumMessageSize = 1024
PrivateKeyPassword = <encrypted>
ServerCertificateFile = /opt/CSCOar/pki/server-cert.pem
ServerKeyFile = /opt/CSCOar/pki/server-key.pem
CACertificateFile = /opt/CSCOar/pki/root-cert.pem
CACertificatePath = /opt/CSCOar/pki
CRLDistributionURL =
ClientVerificationMode = Optional
VerificationDepth = 4
EnableSessionCache = true
UseECCCertificates = true
SessionTimeout = "5 Minutes"
AuthenticationTimeout = 120
```

```
set CACertificateFile PEM:/opt/CSCOar/pki/wimax_device_root.pem
```

```
Set CACertificateFile PEM:/opt/CSCOar/pki/wimax_device_root.pem
```

**Step 3** Save the configuration.

```
save
```

**Step 4** Restart the arserver.

```
/opt/CSCOar/bin/arserver restart
```

---

# Rolling Encryption Support for Pseudonym Generation in EAP-SIM, EAP-AKA, and EAP-AKA' Services

Prime Access Registrar supports rolling encryption which involves generating rolling pseudonym secrets instead of a single pseudonym secret for EAP-SIM, EAP-AKA, and EAP-AKA' services. This feature involves the following objects:

- [User-Defined Keys](#)
- [Key Store](#)
- [EAP Service](#)

## User-Defined Keys

Prime Access Registrar allows you to configure pseudonym encryption keys. There are two types of user-defined keys:

- For generating 3GPP compliant pseudonym secrets—configured at `/Radius/Advanced/KeyStores/3GPPKeys`
- For generating non-3GPP compliant pseudonym secrets—configured at `/Radius/Advanced/KeyStores/non3GPPKeys`

If user-defined keys are not configured, Prime Access Registrar uses system generated keys for rolling encryption.

## Key Store

The Key Store maintains a set of keys that are user-defined and/or auto-generated. There is only one active key at any point in time that is used for generating the pseudonym secret. Rest of the keys are inactive and are used for decryption of old pseudonyms if used for pseudonym generation. After a rollover period, another key is selected as an active key and is used for pseudonym generation. Expiry of a pseudonym depends on the `PseudonymLifetime` property set for the corresponding EAP service.

For each key store, you can configure the number of keys and rollover period for each key, which is one week by default. For each type of EAP service, there are two key stores:

- For generating 3GPP compliant pseudonym secrets—number of keys is limited to 15. Default is 15. Each key should be 16 digits in length.
- For generating non-3GPP compliant pseudonym secrets—number of keys is limited to 1024. Default is 256. Each key should be 29 digits in length.

Following is the CLI configuration of a key store:

```
[ //localhost/Radius/Advanced/KeyStores ]
EAP-FAST/
  NumberOfKeys = 256
  RolloverPeriod = "1 Week"
EAP-SIM/
  NumberOfKeys = 256
  RolloverPeriod = "1 Week"
EAP-SIM-3GPP/
  NumberOfKeys = 15
  RolloverPeriod = "1 Week"
EAP-AKA/
  NumberOfKeys = 256
  RolloverPeriod = "1 Week"
EAP-AKA-3GPP/
  NumberOfKeys = 15
  RolloverPeriod = "1 Week"
```

```

EAP-AKAPRIME/
  NumberOfKeys = 256
  RolloverPeriod = "1 Week"
EAP-AKAPRIME-3GPP/
  NumberOfKeys = 15
  RolloverPeriod = "1 Week"
3GPPKeys/
Non3GPPKeys/

```

#### CLI to configure 3GPP keys:

```

--> cd 3GPPKeys/

[ //localhost/Radius/Advanced/KeyStores/3GPPKeys ]

--> set 1 erafgageasgaafde

Set 1 erafgageasgaafde

--> set 2 tttttttadfadfaff

Set 2 tttttttadfadfaff

```

#### CLI to configure non-3GPP keys:

```

--> cd ../Non3GPPKeys/

[ //localhost/Radius/Advanced/KeyStores/Non3GPPKeys ]

--> set 1 sfsgsgghsr

Set 1 sfsgsgghsr

--> set 2

304 Too few arguments

--> set 2 afgfsfgsf

Set 2 afgfsfgsf

```

### EAP Service

Rolling encryption is applicable for an EAP-SIM, EAP-AKA, or EAP-AKA' service. This feature is applicable only when the parameter `EnableRollingPseudonymSecret` is set to `TRUE` for the service. Each service uses either the 3GPP compliant or non-3GPP compliant pseudonym key store based on the `Generate3GPPCompliantPseudonym` option set for the service.

The EAP service uses the current active key of the corresponding key store to generate the pseudonym. After rollover period of the current active key, the next active key is selected and used for pseudonym generation. When Prime Access Registrar receives the pseudonym back from the client, it decrypts the pseudonym using the same key that was used to generate the pseudonym. Based on the `PseudonymRenewtime` and `PseudonymLifetime` set for the service, if the pseudonym received from the client is expired or to be renewed, Prime Access Registrar uses the current active key from the corresponding key store,

### Rolling Encryption Example

[Table 5-12](#) provides an example of the rolling encryption process.

**Table 5-12 Rolling Encryption Example**

Assumptions	<ul style="list-style-type: none"> <li>• EnableRollingPseudonymSecret and Generate3GPPCompliantPseudonym parameters are set to TRUE.</li> <li>• eap-sim-3gpp key store is used. Current active key in the key store is key x.</li> <li>• Roll over period = 1 week, which means current active key changes every week.</li> <li>• PseudonymRenewtime parameter of the eap service = 2 days, which means the pseudonym is renewed every two days using the current active key.</li> <li>• PseudonymLifetime parameter of the eap service = 5 days, which means pseudonym is expired if is not renewed in five days.</li> </ul>
Day 1	<p>Prime Access Registrar server generates a pseudonym using key x and provides it to the client.</p> <p>Client uses this pseudonym in further authentication process.</p>
Day 2	<p>Client provides the pseudonym back to the Prime Access Registrar server.</p> <p>Prime Access Registrar server decrypts it, identifies the original user, and completes the authentication. No changes are done to the pseudonym.</p>
Day 3	<p>Client contacts Prime Access Registrar server for authentication.</p> <p>Prime Access Registrar understands that the pseudonym needs to be renewed and hence generates a new pseudonym using key x and provides it to the client.</p> <p>Process continues for Days 4 and 5.</p>

**Table 5-12 Rolling Encryption Example (continued)**

Day 'n'	<p>If:</p> <ul style="list-style-type: none"> <li>• there is no authentication transaction between the client and the Prime Access Registrar server for the past five days,</li> <li>• the client already has a pseudonym provided by the Prime Access Registrar server during its last transaction, and</li> <li>• on the 'n'th day, client contacts Prime Access Registrar server for authentication</li> </ul> <p>Then, Prime Access Registrar server:</p> <ul style="list-style-type: none"> <li>• understands that the pseudonym is expired,</li> <li>• requests for a permanent ID,</li> <li>• generates a new pseudonym by using key x, and</li> <li>• provides it to the client.</li> </ul>
Day 'y' (y = rollover period date +1)	<p>Key x changes to 'retired' stage and will be used for decryption of the old pseudonym generated with the old key.</p> <p>Prime Access Registrar server uses the next active key (key x+1) for pseudonym generation.</p> <p>Client can continue the authentication process by using the same old pseudonym till its renewal or expiry time irrespective of active key change at the key store.</p> <p>If the old pseudonym is found to be renewed or expired, Prime Access Registrar server generates the new pseudonym by using key x+1.</p>

## Support for Decrypting Encrypted-IMSI for EAP-SIM, EAP-AKA, and EAP-AKA' Services

Prime Access Registrar supports configuring private keys for decrypting the encrypted-IMSI received from the EAP-client.

An incoming EAP response contains the following components:

- **Delimiter**—Indicates whether the incoming message is encrypted or not. Default value is '\0' (NULL), which indicates that the incoming message contains encrypted IMSI.
- **Permanent Identity**—If the delimiter value is '\0', then the permanent identity in the incoming message is an encrypted IMSI.
- **Key ID Delimiter**—Configured delimiter value, example ',' (comma), that points to a key identifier.
- **Key Identifier**—Helps the server to locate a private key for decryption of the incoming encrypted IMSI.

You can configure the private keys under `/Radius/Advanced/Keystores/EncryptedIMSI-PrivateKeys/keys` for EAP-AKA, EAP-AKA', and EAP-SIM service.

**Note**

The private keys must contain the base64encode padding characters i.e. the final two double equals “==”. Decryption will fail for the keys without “==”.

**Note**

While configuring the private key in Prime Access Registrar, remove the “==” because Prime Access Registrar will add those “==” while processing the request.

A sample configuration of an EAP-AKA service with encrypted IMSI parameters is given below:

```
[ //localhost/Radius/Services/eap-aka ]
  Name = eap-aka
  Description =
  Type = eap-aka
  NumberOfQuintets = 1
  AlwaysRequestIdentity = False
  EnableIdentityPrivacy = False
  EnableRollingPseudonymSecret = False
  PseudonymSecret = <encrypted>
  PseudonymRenewtime = "24 Hours"
  PseudonymLifetime = Forever
  NotificationService =
  Generate3GPPCompliantPseudonym = False
  EnableReauthentication = False
  UseOutagePolicyForReauth = False
  MaximumReauthentications = 16
  ReauthenticationTimeout = 3600
  ReauthenticationRealm =
  EnableEncryptedIMSI = tTRUE
  EncryptedIMSIDelimiter = NULL
  EncryptedIMSIKeyIdDelimiter = ,
  DefaultPrivateKey =
  QuintetCacheTimeout = 120
  AuthenticationTimeout = 120
  QuintetGenerationScript~ =
  UseProtectedResults = False
  SendReAuthIDInAccept = False
  Subscriber_DBLookup =
```

You can configure the private keys under /Radius/Advanced/Keystores/ EncryptedIMSI-PrivateKeys/keys. Sample private key configuration is provided below.

```
--> cd keyStores/

[ //localhost/Radius/Advanced/KeyStores ]
  EAP-FAST/
  EAP-SIM/
  EAP-SIM-3GPP/
  EAP-AKA/
  EAP-AKA-3GPP/
  EAP-AKAPRIME/
  EAP-AKAPRIME-3GPP/
  EncryptedIMSI-PrivateKeys/
  3GPPKeys/
  Non3GPPKeys/

--> cd encryptedIMSI-PrivateKeys/

[ //localhost/Radius/Advanced/KeyStores/EncryptedIMSI-PrivateKeys ]
  AllowedKeyIdentifiers = CertificateSerialNumber
  keys/
```

```

--> cd keys/

[ //localhost/Radius/Advanced/KeyStores/EncryptedIMSI-PrivateKeys/keys ]
  Entries 1 to 1 from 1 total entries
  Current filter: <all>

  keys1/

--> cd keys1/

[ //localhost/Radius/Advanced/KeyStores/EncryptedIMSI-PrivateKeys/keys/keys1 ]
  Name = keys1
  Identifier = CertificateSerialNumber
  Privatekey =

--> set privatekey "ajdosadoiusadosa"

Set Privatekey ajdosadoiusadosa

```

**Note**


---

You must save and reload the server after configuring the private keys for the changes to take effect.

---

## Extended-EAP Support in Prime Access Registrar

Extended-EAP is used as an authorization service to retrieve authorization information from a remote web server using the REST interface. Prime Access Registrar processes all EAP requests, and extends the process through extended EAP service. Extended-EAP service is supported for the following EAP protocols:

- EAP-AKA
- EAP-AKA-Prime
- EAP-SIM

You can configure an extended-EAP service under /Radius/Services. When you define an extended-EAP service under /Radius/Services, you must set the service type to **extended-eap**. Refer to the sample configuration given below:

```

[ //localhost/Radius/Services/extended-EAP ]
  Name = extended-EAP
  Description =
  Type = extended-eap
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = AcceptAll
  OutageScript~ = NASLIST
  NasIDList = NasList
  MultipleServersPolicy = Failover
  RemoteServers/

```

You must also configure a REST remote server for the extended-EAP service. The following is a sample CLI configuration of REST remote server for extended-EAP service:

```

[ //localhost/Radius/RemoteServers/REST-VM035 ]
  Name = REST-VM035
  Description =
  Protocol = rest
  ReactivateTimerInterval = 30000

```

```
Timeout = 1000
MaxTimeOuts = 1
RESTSourceConnections = 1
RequestURL = https://10.81.79.32:8443/eapauth/IMSI/getdetails
UserName = eapAuth32TMUS
Password = <encrypted>
KeepAliveTimerInterval = 1000
RequestToJSONRequestMappings/
  IpAddress = Calling-Station-Id
  nasIdentifier = NAS-Identifier
```

Refer to [REST, page 2-139](#) for details about the REST remote server configuration parameters.





## Using Replication

---

This chapter provides information about how to use the replication feature in Cisco Prime Access Registrar (Prime Access Registrar).

This chapter contains the following sections:

- [Replication Overview](#)
- [How Replication Works](#)
- [Replication Configuration Settings](#)
- [Setting Up Replication](#)
- [Replication Example](#)
- [Full Resynchronization](#)
- [Replication Setup with More Than One Slave](#)



### Note

---

When using replication, use the **aregcmd** command-line interface to make configuration changes to the Prime Access Registrar server. Replication is not supported when using the GUI.

---

## Replication Overview

Prime Access Registrar replication feature can maintain identical configurations on multiple machines simultaneously. When replication is properly configured, changes an administrator makes on the primary or *master* machine are propagated by Prime Access Registrar to a secondary or *slave* machine.

Replication eliminates the need to have administrators with multiple Prime Access Registrar installations make the same configuration changes at each of their installations. Instead, only the master's configuration need be changed and the slave is automatically configured eliminating the need to make repetitive, error-prone configuration changes for each individual installation. In addition to enhancing server configuration management, using replication eliminates the need for a hot-standby machine.

Using a hot-standby machine is a common practice to provide more fault-tolerance where a fully-installed and configured system stands ready to takeover should the primary machine fail. However, a system setup for hot-standby is essentially an idle machine only used when the primary system fails. Hot-standby or secondary servers are expensive resources. Employing Prime Access Registrar's replication feature, both servers can perform RADIUS request processing simultaneously, eliminating wasted resources.

The replication feature focuses on configuration maintenance only, not session information or installation-specific information such as Administrator, Interface, Replication or Advanced machine-specific configuration changes. These configuration items are not replicated because they are specific to each installation and are not likely to be identical between master and slave. While changes to Session Managers, Resource Manager, and Remote Servers are replicated to the slave and stored in the slave's configuration database, they are not hot-configured on the slave (see Hot Configuration Detailed below for more information)

Changes should be made only on the master server. Making changes on a slave server will not be replicated and might result in an unstable configuration on the slave. Any changes made using replication will not be reflected in existing **aregcmd** sessions. **aregcmd** only loads its configuration at start up; it is not dynamically updated. For example, if **aregcmd** is running on the slave, and on the master **aregcmd** is used to add a client, the new client, while correctly replicated and hot-configured, will not be visible in the slave's **aregcmd** until **aregcmd** is exited and restarted.

When there is a configuration change, the master server propagates the change set to all member servers over the network. All member servers have to update their configuration after receiving the change set notifications from master server. Propagating the change set to a member server involves multiple packet transfer from the master server to the member because the master server has to convey all the configuration changes to the member. The number of packets to be transferred depends on the size of the change set.

After receiving a change set notification, the member server will go offline before applying the change set received from master server. This state is indicated by the log message `Radius Server is offline` in **name\_radius\_1\_log** file. When the change set is successfully applied, the member server goes up automatically. This is indicated by the log message `Radius Server is online` in **name\_radius\_1\_log** file. When the member server goes offline to apply the change set, no incoming packets are processed.

Due to the number of packets to be transferred in the change set and the amount of time the member server will be offline updating its database points, we recommend that you use multiple **save** commands rather than a large configuration change with one **save** command. You can also minimize the number of changes that occur in a replication interval by modifying either the `RepTransactionArchiveLimit` or the `RepTransactionSyncInterval`, or both of these properties. For example, instead of using the default value of 100 for the `RepTransactionArchiveLimit`, you might change it to 20.


**Note**


---

The IP address format is enhanced to support both IPv4 and IPv6.

---

## How Replication Works

The following sections describe the flow of a simple replication as it occurs under normal conditions:

- [Replication Data Flow](#)
- [Security](#)
- [Replication Archive](#)
- [Ensuring Data Integrity](#)
- [Full Resynchronization](#)
- [Understanding Hot-Configuration](#)
- [Replication's Impact on Request Processing](#)

## Replication Data Flow

The following sections describe data flow on the master server and the slave server:

- [Master Server](#)
- [Slave Server](#)

### Master Server

The master server or primary server is the fully configured machine that is used to archive all the transactions that taken place in Prime Access Registrar.

#### Performing the Data Flow for the Master Server

To perform data flow for the master server:

- 
- Step 1** The administrator makes a change to the master server's configuration using the **aregcmd** command line interface (CLI) and issues a **save** command.
  - Step 2** After the changes are successfully validated, the changes are stored in the Prime Access Registrar database.
  - Step 3** **aregcmd** then notifies the Prime Access Registrar server executing on the master of the configuration change.
  - Step 4** The Prime Access Registrar server then updates its version of the configuration stored in memory. (This is called *hot-config* because it happens while the server is running and processing requests.)
  - Step 5** The Prime Access Registrar server first copies the changes pertaining to the **aregcmd save**, also known as a transaction to its replication archive, then transmits the transaction to the slave server for processing.
  - Step 6** In **aregcmd**, the prompt returns indicating that the **save** has completed successfully, the transaction has been archived, and the transaction has been transmitted to the slaves.
- 

### Slave Server

The slave server or secondary server is a fully-installed and configured system stands ready to takeover when the primary machine fails.

#### Performing Data Flow for the Slave Server

To perform data flow for the slave server:

- 
- Step 1** When the slave server receives the transaction, its contents are verified.
  - Step 2** After verification, the changes are applied to the slave server's database.
  - Step 3** The changes are then applied (hot-configured) in the slave server's in-memory configuration.
  - Step 4** The transaction is written to the slave server's replication archive.
-

## Security

Replication has two primary security concerns:

1. Security of the transactions transmitted to the slave server
2. Storage of transactions in the replication archive

Both of these concerns use shared secret (MD5) encryption via the shared secret specified in the replication configuration on both master and slave servers. Replication data transmitted between master and slave is encrypted at the source and decrypted at the destination the same way as standard RADIUS packets between Prime Access Registrar's clients and the Prime Access Registrar server. Transactions written to the replication archive are also encrypted in the same manner and decrypted when read from the replication archive.

## Replication Archive

The replication archive serves two primary purposes:

- To provide persistent, or saved, information regarding the last successful transaction
- To persist transactions in case the slave server requires re synchronization (see Ensuring Data Integrity below for more information on re synchronization).

The replication archive is simply a directory located in `../CSCOar/data/archive`. Each transaction replicated by the master is written to this directory as a single file. The name of each transaction file is of the form `txn#####` where `#####` is the unique transaction number assigned by the master server. The replication archive size, that is the number of transaction files it might contain, is configured in the Replication configuration setting of `TransactionArchiveLimit`. When the `TransactionArchive` limit is exceeded, the oldest transaction file is deleted.

## Ensuring Data Integrity

Prime Access Registrar's configuration replication feature ensures data integrity through transaction data verification, transaction ordering, automatic resynchronization and manual full-resynchronization. With the single exception of a manual full-resynchronization, each of the following techniques help to automatically ensure that master and slave servers contain identical configurations. A detailed description of each technique follows. This section contains the following topics:

- [Transaction Data Verification](#)
- [Transaction Order](#)
- [Automatic Resynchronization](#)

### Transaction Data Verification

When the master prepares a transaction for replication to a slave, the master calculates a 2's complement Cyclic Redundancy Check (CRC) for each element (individual configuration change) in the transaction and for the entire transaction and includes these CRC values in the transmitted transaction. When the slave receives the transaction, the slave calculates a CRC for each transaction element and for the entire transaction and compares its own calculated values with those sent with the message. If a discrepancy occurs from these comparisons, the transaction element or the entire transaction is discarded and a re-transmission of that particular transaction element or the entire transaction is requested by the slave from the master. This process is called automatic resynchronization. (described in more detail below)

## Transaction Order

When the master prepares a transaction for replication, it assigns the transaction a unique transaction number. This number is used to ensure the transactions are processed by the slave in exactly the same order as they were processed on the master. Transactions are order dependent. Since the functionality of Prime Access Registrar's configuration replication feature is to maintain identical configurations between master and slave, if transaction order were not retained, master and slave would not contain identical configurations. Consider where two transactions modify the same thing (a defined client's IP address for example). If the first transaction was a mistake and the second was the desired result, the client configuration on the master would contain the second setting; however, if the transactions were processed in the reverse order on the slave, the client configuration on the slave would contain the mistaken IP Address. This example illustrates the critical need for transaction ordering to ensure data integrity.

## Automatic Resynchronization

Automatic Resynchronization is the most significant feature with respect to data integrity. This feature ensures the configurations on both the master and slave are identical. If they are not, this feature automatically corrects the problem.

When the master and slave start-up, they determine the transaction number of the last replication transaction from their respective replication archives. The master immediately begins periodic transmission of a TransactionSync message to the slave. This message informs the slave of the transaction number of the transaction that the master last replicated.

If the transaction number in the TransactionSync message does not match the transaction number of the last received transaction in the slave's archive, then the slave will request resynchronization from the master. The resynchronization request sent by the slave will include the slave's last received transaction number.

The master will respond by retransmitting each transaction since the last transaction number indicated by the slave in the resynchronization request. The master obtains these transactions from its replication archive.

Should the slave's last received transaction number be less than the lowest transaction number in the master's replication archive, then automatic resynchronization cannot occur as the master's replication archive does not contain enough history to synchronize the slave. In this case, the slave must be resynchronized with a full-resynchronization.

## Full Resynchronization

Full Resynchronization means that the slave has missed more transactions than are stored in the master's replication archive and cannot be resynchronized automatically. There is no automatic full-resynchronization mechanism in Prime Access Registrar's configuration replication feature. To perform a full resynchronization, see [Full Resynchronization](#).

## Understanding Hot-Configuration

Hot-Configuration is the process of reflecting configuration changes made to Prime Access Registrar's internal configuration database in the in-memory configuration of the executing Prime Access Registrar server. Hot-Configuration is accomplished without interruption of RADIUS request processing. For example, if an administrator uses **aregcmd** to configure a new client and issues a **save** command, when the prompt returns, the newly configured client can send requests to Prime Access Registrar.

Hot-Configuration minimizes the down-time associated with having to restart an Prime Access Registrar server to put configuration changes into effect. With the Hot-Configuration feature, a restart is only necessary when a Session Manager, Resource Manager or Remote Server configuration is modified. These configuration elements might not be hot-configured because they maintain state (an active session, for example) and cannot be modified without losing the state information they maintain. Changes to these configuration elements require a restart of Prime Access Registrar to put them into effect.

Hot-Configuration's only connection to the replication feature is that when a change is replicated to the slave, the slave is hot-configured to reflect the replicated change as if an administrator had used **aregcmd** to make the changes directly on the slave server.

**Note**

---

Any misconfiguration of replication will impact the performance of Prime Access Registrar. Hence manual resync is required to rectify the misconfiguration.

---

## Replication's Impact on Request Processing

The replication feature was designed to perform replication of transactions with minimal impact on RADIUS request processing. When a transaction is received by a slave, RADIUS requests are queued while the transaction is applied to the slave. After the transaction is complete, RADIUS request processing resumes.

The impact on RADIUS request processing is a direct result of the size of a transaction. The smaller the transaction the lesser the impact, and the larger the transaction, the greater the impact. In other words, when making changes to the master, frequent saves are better than making lots of changes and then saving. Each change is one transaction element and all changes involved in a **save** comprise a single transaction with one element per change. Since the replication feature only impacts RADIUS request processing when changes are made, the impact under normal operation (when changes are not being made) is virtually unmeasurable.

## Replication Configuration Settings

This section describes each replication configuration setting. In **aregcmd**, replication settings are found in **//localhost/Radius/Replication**. This section contains the following topics:

- [RepType](#)
- [RepTransactionSyncInterval](#)
- [RepTransactionArchiveLimit](#)
- [RepIPAddress](#)
- [RepPort](#)
- [RepSecret](#)

- [RepIsMaster](#)
- [RepMasterIPAddress](#)
- [RepMasterPort](#)
- [Rep Members Subdirectory](#)
- [Rep Members/Slave1](#)
- [Name](#)
- [IPAddress](#)
- [Port](#)

## RepType

RepType indicates the type of replication. The choices available are SMDBR and NONE.

When RepType is set to NONE, replication is disabled. To enable replication, set RepType to SMDBR for Single Master DataBase Replication. RepType must be set to SMDBR on both the master and slave servers.

## RepTransactionSyncInterval

### Master

On the master server, RepTransactionSyncInterval is the duration between periodic transmission of the TransactionSync message expressed in milliseconds. The default is 60000 or 1 minute.

The purpose of RepTransactionSyncInterval is to indicate how frequently to check for an out-of-sync condition between the master and slave servers. When the slave received the TransactionSync message, it uses its contents to determine if it needs to resynchronize with the master.

The larger the setting for RepTransactionSyncInterval, the longer the period of time between out-of-sync detection. However, if RepTransactionSyncInterval is set too small, the slave can frequently request resynchronization when it is not really out of sync. If the duration is too small, the slave cannot completely receive a transaction before it receives the TransactionSync message. In this case, the servers will remain synchronized, but there will be unnecessary excess traffic that could affect performance.

**Note**

---

We recommend that you use smaller values for the RepTransactionSyncInterval to limit the time a slave server is offline applying change sets during automatic resynchronization.

---

### Slave

On the slave, RepTransactionSyncInterval is used to determine if the slave has lost contact with the master and to alert administrators of a possible loss of connectivity between the master and slave. If the elapsed time since the last received TransactionSync message exceeds the setting of RepTransactionSyncInterval, the slave writes a log message indicating that it might have lost contact with the master. This log message is repeated each TransactionSyncInterval until a TransactionSync message is received.

## RepTransactionArchiveLimit

On both master and slave, the RepTransactionArchiveLimit setting determines how many transactions can be stored in the archive. The default setting is 100. When the limit is exceeded, the oldest transaction file is deleted. If a slave requires resynchronization and the last transaction it received is no longer in the archive, a full resynchronization will be necessary to bring the slave back in sync with the master.

**Note**

---

The value set for RepTransactionArchiveLimit should be the same on the master and the slave.

---

An appropriate value for RepTransactionArchiveLimit depends upon how much hard disk space an administrator can provide for resynchronization. If this value is large, say 10,000, then the last 10,000 transactions will be stored in the archive. This is like saying the last 10,000 saves from **aregcmd** will be stored in the archive. Large values are best. The size of each transaction depends upon how many configuration changes were included in the transaction, so hard disk space usage is difficult to estimate.

**Note**

---

We recommend that you use smaller values for the RepTransactionArchiveLimit to limit the time a slave server is offline applying change sets during automatic resynchronization.

---

If the slave should go down or otherwise be taken off line, the value of RepTransactionArchiveLimit and the frequency of **aregcmd** saves will determine how long the slave can be offline before a full-resynchronization will be required.

There are two reasons why a slave server should have an archive:

1. The slave must save the last received transaction for resynchronization purposes (at a minimum).
2. Should the master go down, the slave can then be configured as the master and provide resynchronization services to other slaves.

## RepIPAddress

The RepIPAddress value is set to the IP Address of the machine containing the Prime Access Registrar installation.

**Note**

---

The IP address format is enhanced to support both IPv4 and IPv6.

---

## RepPort

The RepPort is the port used to receive of replication messages. In most cases, the default value (1812) is sufficient. If another port is to be used, the interfaces must exist in the machine.

## RepSecret

RepSecret is the replication secret shared between the master and slave. The value of this setting must be identical on both the master and the slave.

## RepIsMaster

The RepIsMaster setting indicates whether the machine is a master or a slave. On the master, set RepIsMaster to TRUE. On the slave set it to FALSE. Only the master can have this value set to TRUE and there can be only one master.

## RepMasterIPAddress

RepMasterIPAddress specifies the IP Address of the master. On the master, set RepMasterIPAddress to the same value used in RepIPAddress above. On the slave, RepMasterIPAddress must be set to the IP Address of the master.

**Note**

---

The IP address format is enhanced to support both IPv4 and IPv6.

---

## RepMasterPort

RepMasterPort is the port to use to send replication messages to the master. In most cases, the default value (1812) is sufficient; however, if another is to be used, the interfaces must exist in the machine.

## Rep Members Subdirectory

The Rep **Members**\ subdirectory contains the list of slaves to which the master will replicate transactions.

## Rep Members/Slave1

Each slave is added much like a client is added. Each slave must have a configuration in the Rep Members directory to be considered part of the *replication network* by the master. The master will not transmit any messages or replications to servers not in this list, and any communication received by a server not in this list will be ignored.

**Note**

---

Although it is possible to configure multiple slaves with the same master, we have only considered a single-master/single-slave configuration. This is the recommended configuration.

---

## Name

This is the name of the slave. The name must be unique.

## IPAddress

This is the IP Address of the slave.

**Note**

The IP address format is enhanced to support both IPv4 and IPv6.

## Port

This is the port upon which the master will send replication messages to the slave.

## Setting Up Replication

This section provides step-by-step instructions about how to configure replication on both the master and member servers. The “[Replication Example](#)” section on page 6-13, shows an example of replication configuration.

If possible, open an **xterm** window on both the master and member. In each of these windows, change directory to **\$INSTALL/logs** and run **xtail** to watch the logs. This allows you to watch replication log messages as they occur. If you are using a system which had a previous installation of Prime Access Registrar, delete all files located in the **\$INSTALL/data/archive** directory if it is present on either the master or member systems. This section contains the following topics:

- [Configuring The Master](#)
- [Configuring The Member](#)
- [Verifying the Configuration](#)

## Configuring The Master

On the master server, **RepTransactionSyncInterval** is the duration between periodic transmission of the **TransactionSync** message expressed in milliseconds. The default is 60000 or 1 minute.

### Configuring the Master Server for Replication

To configure the master server for replication:

- 
- Step 1** On the machine which is to be the master, using **aregcmd**, navigate to **//localhost/Radius/Replication**
- Step 2** Set the **RepType** to **SMDBR**:
- ```
set RepType SMDBR
```
- Step 3** Set the **RepIPAddress** to the IP address of the master:
- ```
set RepIPAddress 192.168.1.1
```
- Step 4** Set the **RepSecret** to **MySecret**:
- ```
set RepSecret MySecret
```
- Step 5** Set **RepIsMaster** to **TRUE**:
- ```
set RepIsMaster TRUE
```

**Step 6** Set RepMasterIPAddress to the same value used in step 3:

```
set RepMasterIPAddress 192.168.1.1
```

**Step 7** Change directory to **/Radius/Advanced** and set the **MaximumNumberOfRadiusPackets** property to 8192:

```
cd /Radius/Advanced
```

```
set MaximumNumberOfRadiusPackets 8192
```

**Step 8** Change directory to **Rep Members**:

```
cd "rep members"
```



---

**Note** You must enclose Rep Members in quotes due to the space in the name.

---

**Step 9** Add **member1**:

```
add member1
```

**Step 10** Change directory to **member1**:

```
cd member1
```

**Step 11** Set the IPAddress to the IP Address of the machine to be the member:

```
set IPAddress 192.168.1.2
```



---

**Note** The RepPort and RepMasterPort properties on the Master must correspond to one of the ports configured in **/Radius/Advanced/Ports**, if one is configured. Otherwise, the default values for the RepPort and RepMasterPort properties are sufficient.

---

**Step 12** Save the configuration:

```
save
```

**Step 13** Reload the configuration:

```
reload
```

---

## Configuring The Member

On the slave, RepTransactionSyncInterval is used to determine if the slave has lost contact with the master and to alert administrators of a possible loss of connectivity between the master and slave.

### Configuring the Member Server for Replication

To configure the member server for replication:

---

**Step 1** On the machine which is to be the member, using **aregcmd**, navigate to **//localhost/Radius/Replication**.

**Step 2** Set the RepType to SMDBR.

```
set RepType SMDBR
```

**Step 3** Set the RepIPAddress to the IP address of the member.

```
set RepIPAddress 192.168.1.2
```

**Step 4** Set the RepSecret to MySecret.

```
set RepSecret MySecret
```

**Step 5** Set RepMasterIPAddress to IP Address of the master (the same value used in Step 3 on page 8-1).

```
set RepMasterIPAddress 192.168.1.1
```

**Step 6** Change directory to **/Radius/Advanced** and set the **MaximumNumberOfRadiusPackets** property to 8192.

```
cd /Radius/Advanced
```

```
set MaximumNumberOfRadiusPackets 8192
```

**Step 7** If the Master has been configured to use a port other than the well-known (and default) RADIUS ports, configure each Member to use the same port.



---

**Note** The RepPort and RepMasterPort properties on the Master must correspond to one of the ports configured in **/Radius/Advanced/Ports**, if one is configured. Otherwise, the default values for the RepPort and RepMasterPort properties are sufficient.

---

**Step 8** Save the configuration:

```
save
```

**Step 9** Reload the configuration:

```
reload
```

---

## Verifying the Configuration

After both servers have successfully started, use **aregcmd** to make a small change to be replicated to the member server which you can easily verify. We recommend setting the description in **//localhost/Radius** to something like *Test1*. After you issue an **aregcmd save** and the prompt returns, run **aregcmd** on the member server and change directory to **//localhost/Radius**. Ensure that the description is set to Test1. If this was successful, then replication is properly configured and functional.

# Replication Example

This section provides an example of replication and shows the actions that occur.

## Adding a User

The **Users** object contains all of the information necessary to authenticate a user or authorize a user. Users in local UserLists can have multiple profiles. On the master server, use **aregcmd** to add a new user to the default user list.

### Adding a New User

To add a new user:

- 
- Step 1** Change directory to `//localhost/Radius/UserLists/Default`.
- Step 2** Enter the following:
- ```
add testuser
```
- Step 3** Change directory to `testuser`.
- ```
cd testuser
```
- Step 4** Set the password for `testuser`.
- ```
set password testuser
```
- Step 5** Confirm the password by entering `testuser` again.
- Step 6** Enter `save` to save the configuration.
- 

## Master Server's Log

The log on the master shows the following:

```
*** ./name_radius_1_log ***
10/23/2013 23:17:07 name/radius/1 Info Server 0 Initiating Replication of Transaction
1 with 2 Elements.
10/23/2013 23:17:07 name/radius/1 Info Server 0 Replication Transaction #1 With 2
Elements Initiated
```

## Member Server's Log

The log on the member shows the following:

```
*** ./name_radius_1_log ***
10/23/2013 23:15:18 name/radius/1 Info Server 0 Radius Server is On-Line
10/23/2013 23:17:12 name/radius/1 Info Server 0 Committing Replication of Transaction
1 with 2 Elements.
10/23/2013 23:17:16 name/radius/1 Info Server 0 Replication Transaction #1 With 2
Elements Committed.
```

## Verifying Replication

You can use one of two methods to verify that the new user `testuser` was properly replicated to the member:

- Run **aregcmd** on the member and look at the default `userlist` to see if it is there.
- Run **radclient** on the member and enter **simple testuser testuser** to create a simple access request packet (p001).

Enter **p001 send** to send it. When it returns with p002, enter **p002** to see if it is an Access Accept packet or an Access Reject packet. If it is an Access Accept, the user was properly replicated to the member. Using **radclient** is the recommended method to validate that a user was properly replicated.

On the Master, use **aregcmd** to delete the user from the default user list and save the user list.

## Master Server's Log

The log on the master shows the following:

```
*** ./name_radius_1_log ***
10/23/2013 23:20:48 name/radius/1 Info Server 0 Initiating Replication of Transaction
2 with 1 Elements.
10/23/2013 23:20:48 name/radius/1 Info Server 0 Replication Transaction #2 With 1
Elements Initiated
```

## Member Server's Log

The log on the member shows the following:

```
*** ./name_radius_1_log ***
10/23/2013 23:20:53 name/radius/1 Info Server 0 Committing Replication of Transaction
2 with 1 Elements.
10/23/2013 23:20:57 name/radius/1 Info Server 0 Replication Transaction #2 With 1
Elements Committed.
```

Repeat the validation procedure above to ensure the user `testuser` is no longer present on the member.

## Using aregcmd -pf Option

Prime Access Registrar's replication feature works well using **aregcmd** input files. An **aregcmd** input file contains a list of **aregcmd** commands. For example, if the initial configuration of Prime Access Registrar were constructed in an input file, the master and member could be configured for replication first, then the input file applied to the master will be automatically replicated to the member.

### Using aregcmd -pf Option

To illustrate replication using an **aregcmd** input file:

---

**Step 1** Create a text file called **add5users** with the following commands:

```
add /Radius/UserLists/Default/testuser1
cd /Radius/UserLists/Default/testuser1
```

```
set password testuser1
add /Radius/UserLists/Default/testuser2
cd /Radius/UserLists/Default/testuser2
set password testuser2
add /Radius/UserLists/Default/testuser3
cd /Radius/UserLists/Default/testuser3
set password testuser3
add /Radius/UserLists/Default/testuser4
cd /Radius/UserLists/Default/testuser4
set password testuser4
add /Radius/UserLists/Default/testuser5
cd /Radius/UserLists/Default/testuser5
set password testuser5
save
```

**Step 2** On the master server, run the following command:

```
aregcmd -pf add5users
```

---

## Master Server's Log

The log on the master shows the following:

```
*** ./name_radius_1_log ***
10/23/2013 23:27:08 name/radius/1 Info Server 0 Initiating Replication of Transaction
3 with 10 Elements.
10/23/2013 23:27:08 name/radius/1 Info Server 0 Replication Transaction #3 With 10
Elements Initiated
```

## Member Server's Log

The log on the member shows the following:

```
*** ./name_radius_1_log ***
10/23/2013 23:27:12 name/radius/1 Info Server 0 Committing Replication of Transaction
3 with 10 Elements.
10/23/2013 23:27:17 name/radius/1 Info Server 0 Replication Transaction #3 With 10
Elements Committed.
```

When the prompt returns, go to the member and use **aregcmd** to view the **/radius/defaults/userlist**. There should be five users there named *testuser1* through *testuser5*.

## An Automatic Resynchronization Example

This example will illustrate resynchronization of the member. This will be accomplished by stopping the member, making changes on the master, then restarting the member forcing a resynchronization.

### Performing Resynchronization of the Member

To perform resynchronization of the member:

---

**Step 1** At the member, stop the Prime Access Registrar server:

```
/etc/init.d/arservagt stop
```

At the master, run **aregcmd** and change directory to **/radius/userlist/default**.

```
cd /radius/userlist/default
```

**Step 2** Enter the following:

```
add foouser
```

**Step 3** Change directory to **foouser**.

```
cd foouser
```

**Step 4** Set the password for **foouser**.

```
set password foouser
```

**Step 5** Confirm the password by entering **foouser** again.

**Step 6** Save the configuration:

```
save
```

---

## Master Server's Log

The log on the master shows the following:

```
*** ./name_radius_1_log ***
10/23/2013 23:31:02 name/radius/1 Info Server 0 Initiating Replication of Transaction
5 with 2 Elements.
10/23/2013 23:31:02 name/radius/1 Info Server 0 Replication Transaction #5 With2
Elements Initiated
```

On the member, run **/etc/init.d/arservagt start**. Notice the following log messages in the Master's log:

```
*** ./name_radius_1_log ***
10/23/2013 23:33:19 name/radius/1 Info Server 0 Resynchronizing member1.
```

## Member Server's Log

The log on the member shows the following:

```
*** ./name_radius_1_log ***
11/07/2013 23:33:14 name/radius/1 Info Server 0 Radius Server is Off-Line
11/07/2013 23:33:14 name/radius/1 Info Server 0 Starting Replication Manager
11/07/2013 23:33:24 name/radius/1 Info Server 0 Master Selected As Partner (DEFAULT)
11/07/2013 23:33:24 name/radius/1 Info Server 0 Radius Server is Off-Line
11/07/2013 23:33:24 name/radius/1 Warning Server 0 Requesting resynchronization from
Master: Last Txn#3
11/07/2013 23:33:24 name/radius/1 Info Server 0 Resynchronization from Master in
progress.
11/07/2013 23:33:24 name/radius/1 Info Server 0 Committing Replication of Transaction
4 with 2 Elements.
11/07/2013 23:33:28 name/radius/1 Info Server 0 Replication Transaction #4 With 2
Elements Committed.
11/07/2013 23:33:28 name/radius/1 Info Server 0 Radius Server is On-Line
```

As the log above shows, when the member started up, it validated its last received transaction number (#3) with the master's last replicated transaction number (#4). They did not match because a replication was initiated by the master which was not received by the member (because the member was stopped). When the member detected this discrepancy, the member made a resynchronization request to the master. The master responded by transmitting the missed transaction (#4) to the member. After it received and processed the retransmitted transaction, the member determined that it was then synchronized with the master and placed itself in an online status.

## Full Resynchronization

Full Resynchronization means that the member has missed more transactions than are stored in the master's replication archive and can not be resynchronized automatically. There is no automatic full-resynchronization mechanism in Prime Access Registrar's configuration replication feature. If a full resynchronization is required, you must export the master server's database and update the member configuration.



### Note

Before beginning, ensure there are no **aregcmd** sessions logged into the master server.

### Performing a Manual Full-resynchronization

To perform a manual full-resynchronization:

- 
- Step 1** On the master server, stop the Prime Access Registrar server agent using the following command:  
`/etc/init.d/arserver stop`
  - Step 2** On the master server, change directory to `$INSTALL/data/db`.
  - Step 3** Create a tarfile made up of the three database files, `mcddb.d01`, `mcddb.d02`, and `mcddb.d03`.  
`tar cvf /tmp/db.tar mcddb.d0*`
  - Step 4** Create a tarfile of the archive.  
`tar cvf /tmp/archive.tar $INSTALL/data/archive`
  - Step 5** On the master server, start the Prime Access Registrar server agent using the following command:

**/etc/init.d/arserver start**

**Step 6** On each member server requiring resynchronization, perform the following:

- a. On the member server, stop the Prime Access Registrar server agent using the following command:

**/etc/init.d/arserver stop**

- b. Copy the tarfiles (**db.tar** and **archive.tar**) to **/tmp**.

- c. Change directory to **\$INSTALL/data/db**, then untar the compressed database files.

**cd \$INSTALL/data/db**

**tar xvf /tmp/db.tar**

- d. Rebuild the key files using the following command:

**\$INSTALL/bin/keybuild mcddb**




---

**Note** This step might take several minutes.

---

- e. Untar the archive.

**cd \$INSTALL/data/archive**

**tar xvf /tmp/archive.tar**

- f. As a safety check, run the following UNIX command to verify the integrity of the database.

**\$INSTALL/bin/dbcheck mcddb**




---

**Note** You must be user **root** to run **dbcheck**.

---

No errors should be detected.

- g. Start the Prime Access Registrar server agent using the following command:

**/etc/init.d/arserver start**




---

**Note** After you start the member server with the master server's database, you will probably see messages such as the following:

---

```
11/07/2013 23:21:23 name/radius/1 Error Server 0 TXN_SYNC: Failed to get master's
socket handle.
```

```
11/07/2013 23:21:49 name/radius/1 Warning Server 0 TXN_SYNC Received by Master from
unknown member 10.1.9.74. Validation Failed
```

These messages will likely continue until you complete steps **h** and **i**.

---

- h. Change directory to **//radius/replication** and change the following attributes:

- Change the RepIPAddress to that of the member.
- Change RepIsMaster to FALSE.

- Remove any entries under Rep Members.
- i. Save and reload the configuration.

**save**

```
Validating //localhost...  
Saving //localhost...
```

**reload**

The member will start up and show online status in the log after it has verified it is synchronized with the master.

---

## Replication Setup with More Than One Slave

When replication is set up with more than one slave, Prime Access Registrar's replication feature ensures that all the servers maintain identical configuration. This is done by forming a communication mesh. This mesh is formed by every server choosing two partners for itself from the replication setup. The servers tend to receive/send configuration updates from/to its partners. This ensures that all the servers maintain identical configuration inspite of minimal communication failures.

When bringing up a replication setup, Prime Access Registrar server comes up first and then initiates a partner sync request to all its replication members as visible from the configuration. So, a slave server will initiate partner sync to its master only. This is because master server is the only server visible to the slave server from the configuration. The master server will then broadcast the partner syncs that it has received, to all its replication members (slaves). Based on the sync messages sent by the master to this server, the evaluation of workload happens. The partner selection is based on the workload evaluation. Choosing the partners based on workload, ensures that the workload is equally distributed across the partner network.

The partners are selected based on the count of partner syncs received from the master:

- If partner syncs that have been received is one, choose the master as a partner.
- If partner syncs that have been received is two, choose the master and the other replication server as partners.
- If partner syncs that have been received is greater than two, perform a workload evaluation on the partners. Identify two servers that do not have two partners and choose them as partners.





## Using Identity Caching

Cisco Prime Access Registrar (Prime Access Registrar) software includes the identity caching feature. Prime Access Registrar runs as application layer software and can be used standalone or in conjunction with other workstations running Prime Access Registrar.



**Note**

The identity caching feature is available on Prime Access Registrar releases 3.5.2 and above.

Identity caching provides subscriber identity resolution services with fast access to associated subscriber identity data for service providers, enabling them to offer new services to their customers based on identity caching and context information management.

This chapter contains the following sections:

- [Overview](#)
- [Identity Caching Features](#)
- [Configuring Cisco Prime Access Registrar for Identity Caching](#)
- [Starting Identity Caching](#)

## Overview

Identity caching enables Cisco equipment to gain context information about the operator's subscribers to support network functions or to enhance subscriber's experience on the operator's network. [Figure 7-1 on page 7-2](#), Prime Access Registrar System Overview, shows the network environment where Prime Access Registrar identity caching might be used.

For example, Client Services Gateway (CSG) uses IP mapping information provided by identity caching to support post-paid content billing. Identity caching acquires subscriber information from other devices and information sources in the operator's network. The type of information provided is limited by the available information sources and is configurable by the operator, but might include information such as IP address, MSISDN, and IMSI. Identity caching does not duplicate the operator's persistent data stores. Identity caching provides a protocol-based interface through which Cisco network elements (Prime Access Registrar identity caching clients) can access subscriber information.

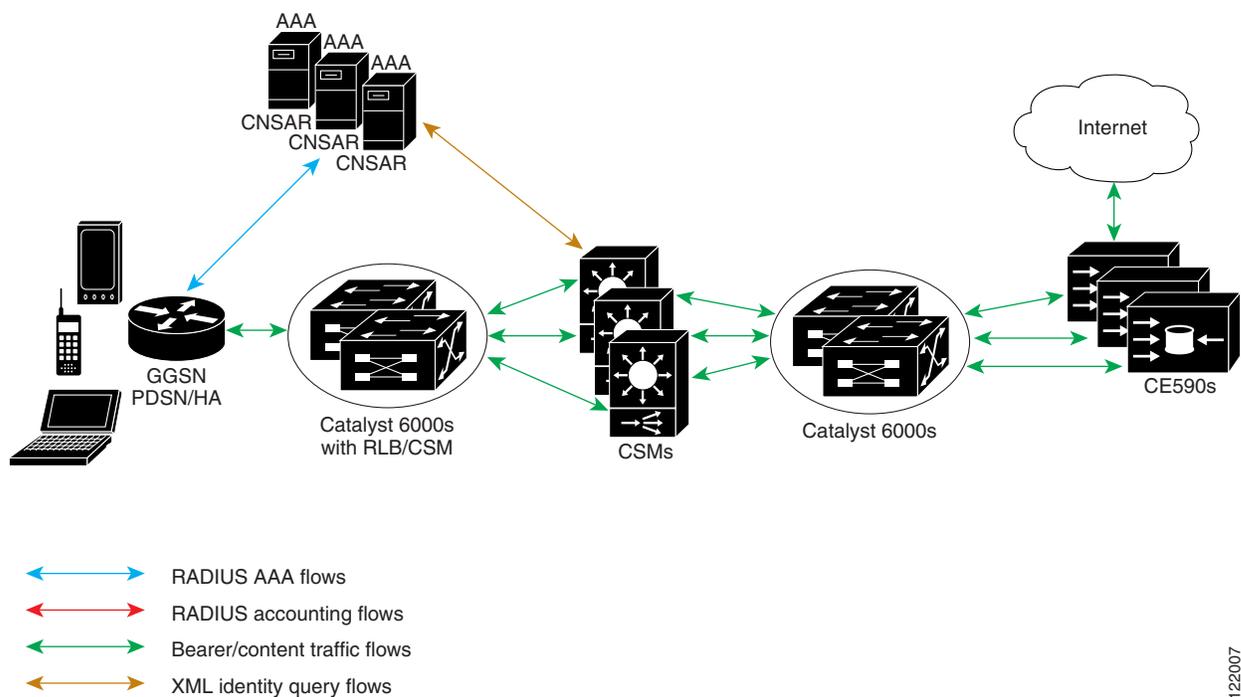
The Prime Access Registrar servers receive RADIUS flows from the Gateway GPRS support Node (GGSN) which acts as a type of network access station (NAS). These flows perform full AAA (authentication, authorization, and accounting). You can configure the Prime Access Registrar servers to redirect the accounting information (only) to an identity caching server to be cached. The GGSN can also be configured to direct only the RADIUS accounting information directly to the Prime Access Registrar server.

Prime Access Registrar also receives XML identity query flows from the CSM which acts as a NAS. In the event that a CSM should fail or lose its information, the information can be refreshed from the information cached in the Prime Access Registrar server.

Prime Access Registrar acquires subscriber information such as the IP address, the mobile Subscriber ISDN number (MSISDN), and the International Mobile System Identifier (IMSI) from AAA requests the Prime Access Registrar server receives, typically from the GGSN. The types of information provided is limited by the available information sources and is configurable by the operator.

Prime Access Registrar includes an XML Query Identity enhancement. Prime Access Registrar previously supported User-Name lookup based on the Framed IP address of an existing session. The XML Query Identity enhancement enables Framed IP address lookup based on the User-Name in an existing session.

**Figure 7-1 Prime Access Registrar System Overview**



122007

## Identity Caching Features

Prime Access Registrar identity caching provides the following features:

- Supports GGSN subscriber data attributes from RADIUS authentication sequences
- Provides basic identity mapping services from IP address or username/APN to Mobile DN for one network presence at a time.
- Provide session management support for Content Switch Module (CSM)

Prime Access Registrar enables the CSM to keep the data session and content correlated to the same subscriber reconnecting, perhaps after an attach/detach sequence for a GPRS subscriber connecting again. This is done through the MSISDN identity to IP mapping in the identity caching function.

- Enhance redundancy with stateful fail-over support for applications by finding the right connection between subscriber identity and IP address using the Identity Cache function.
- Uses an XML interface to make it easier for any network function or application to use without having to have detailed internal knowledge about the execution environment or programming methods.
- Provides user identity resolution with fast access to associated subscriber data
- Establishes an identity and Access Management solution that can be used in and across multiple network domains
- Provides a way to use identity resolution to manage the growth of 2.5G mobile data access services (GSM/GPRS) and to provide always-on mobile data access including the following:
  - Ties various IP addresses to a unique subscriber identifier
  - Dynamically assigning and reusing IP addresses and controlling services with consistent identification
  - Correlates previous content activity when a mobile subscriber reconnects
  - Correlates IP addresses, mobile numbers, username, and identifiers to support customer billing
  - Correlates and identifies subscribers using both 2.5G and WLAN services and provides a way to control and manage operator network services
  - Provides subscriber privacy control
  - Provides a way to cache content with various customers and their networks

## Configuring Cisco Prime Access Registrar for Identity Caching

Use the command line interface **aregcmd** to configure Prime Access Registrar to perform identity caching.

### Configuring the Identity Caching

To configure identity caching:

- 
- Step 1** Launch **aregcmd**.
- Step 2** Define a client object for each client that will send either RADIUS or XML packets to the Prime Access Registrar server performing identity caching.

There should be one client object for each GGSN, one for each CSM and one for each packet simulator (if used in a test environment).

For example, if a packet simulator will be used on the same server where you perform identity caching, add a client object as in the following:

```
cd /Radius/Clients
add xml-client
cd xml-client
```

```
[ //localhost/Radius/Clients/xml-client ]
Name = xml-client
Description =
IPAddress =
SharedSecret =
Type = NAS
Vendor =
IncomingScript~ =
OutgoingScript~ =
EnablePOD = FALSE
```

This client object is very similar to the localhost object defined in the example configuration. The **SharedSecret** property will be ignored if the client is an XML client, but still must be set to a non-null value. The **Type** property is also ignored for XML clients.

- Step 3** Define a port object for each RADIUS port and each XML port to be used. Two RADIUS ports, the second immediately following the first in numeric value, must be defined even if only one is needed. A typical identity caching installation requires the following port configuration:

```
cd /Radius/Advanced/Ports

add 1812

add 1813

add 8080
```




---

**Note** Although ports 1812 and 1813 are the default ports for Prime Access Registrar, you must add them to **/Radius/Advanced/Ports** to also add port 8080.

---

- Step 4** Change directory to the 1812 port and set its type to Radius-Access.

```
cd /Radius/Advanced/Ports/1812

set Type Radius-Access
```

- Step 5** Change directory to the 1813 port and set its type to Radius-Accounting.

```
cd /Radius/Advanced/Ports/1813

set Type Radius-Accounting
```

- Step 6** Change directory to the 8080 port and set its type to XML.

```
-cd /Radius/Advanced/Ports/8080

set Type XML
```

- Step 7** Define and configure an accounting service of type file and set it as the DefaultAccountingService.

An accounting service is required for Prime Access Registrar to cache identity information, even if no accounting service is needed otherwise. If you added the example configuration during installation, a local-file accounting service is already configured.

If you did not add the example configuration during software installation, see the [Setting Up Accounting](#) section in [Chapter 3, “RADIUS Accounting.”](#)

**Step 8** Define and configure a ResourceManager for identity caching.

```
cd /Radius/ResourceManagers
add cache
```

**Step 9** Set the ResourceManager to type session-cache for identity caching.

```
cd cache
set type session-cache
```

The following shows the default properties of a session-cache ResourceManager:

```
[ //localhost/Radius/ResourceManagers/cache ]
Name = cache
Description =
Type = session-cache
OverwriteAttributes = FALSE
QueryKey =
PendingRemovalDelay = 10
AttributesToBeCached/
QueryMappings/
```

**Step 10** Set the QueryKey to a RADIUS attribute you want to key on.

For example, use the following command to set the QueryKey to User-Name:

```
set QueryKey User-Name
```

The QueryKey must match the string on the right-hand side of one of the pairs you list in QueryMappings. It is not necessary for the QueryKey to be configured under **AttributesToBeCached** because the QueryKey will always be cached by default.



**Note**

The QueryKey property must always be a RADIUS attribute. The Prime Access Registrar server forces a NULL IP address (0.0.0.0) if it detects an incorrectly configured QueryKey.

**Step 11** Change directory to **AttributesToBeCached** and use the **set** command to provide a list of RADIUS attributes you want to store in cache.

```
cd AttributesToBeCached
set 1 Calling-Station-ID
Set 2 User-Name
Set 3 Framed-IP-Address
```

The attributes a session-cache resource manager caches can be queried through both RADIUS Query and XML Query packets. When you cache attributes Framed-IP-Address or User-Name, or when you use XML-Address-format-IPv4 or XML-UserId-id\_type-subscriber\_id as the QueryKey, you must map the XML attributes to RADIUS attributes in the **QueryMappings** subdirectory.

**Step 12** Change directory to **QueryMappings** and use the **set** command to list the attribute pairs, mapping the XML attributes on the left-hand side to the RADIUS attribute on the right-hand side.

```
set XML-Address-format-IPv4 Framed-IP-Address
```

```
set XML-UserId-id_type-subscriber_id User-Name
```

**Step 13** Change directory to **/Radius/SessionManagers** and add a SessionManager for identity caching.

```
cd /Radius/SessionManagers
```

```
add IDcache
```

**Step 14** Change directory to the new identity caching SessionManager, then change directory to the **ResourceManager** list.

```
cd IDcache/ResourceManagers
```

**Step 15** Use the **set** command to associate the identity caching ResourceManager with this SessionManager.

```
set 1 cache
```

**Step 16** Change directory to **/Radius** and set the DefaultSessionManager to the identity caching SessionManager.

```
cd /Radius
```

```
set DefaultSessionManager IDcache
```

**Step 17** Run the **save**, **reload**, and **exit** commands:

```
save
```

```
reload
```

```
exit
```

## Starting Identity Caching

To start identity caching, you must send an Accounting-Request to the specified accounting port (The default accounting port is 1813.) A minimal Accounting-Request will contain the following attributes:

- NAS-Identifier or NAS-IP-Address
- NAS-Port
- Framed-IP-Address
- User-Name
- Acct-Status-Type
- Acct-Session-Id

### Starting Identity Caching

To start identity caching:

---

**Step 1** Launch **radclient**:

```
cd /opt/CSCOAr/bin  
  
radclient -C localhost -N admin -P aicuser
```

**Step 2** Enter the following **radclient** commands:

```
set p [ acct_request Start joeuser@cisco.com ]  
  
$p set attrib [ attrib Framed-IP-Address 123.123.123.123 ]  
  
$p send
```

This assumes that you are running **radclient** on the same server and using 1813 as the accounting port.

**Step 3** Send XML requests to the specified XML port (Cisco suggests port 8080 as shown above). A typical XML packet will look like the following:

```
<?xml version="1.0"?>  
<Request>  
  <UserIdRequest>  
    <UserId id_type="subscriber_id">bob</UserId>  
  </UserIdRequest>  
</Request>
```

To do this using **xmlclient**, put the XML text into a file, then enter the following command:

```
cd /opt/CSCOAr/bin  
  
./xmlclient -srd <file>
```

**Note**

This assumes that **xmlclient** is running on the same server as identity caching and that 8080 is the XML port. Use the command **xmlclient -H** for information about how to use a different port or how to run **xmlclient** from a different server.

**Note**

For a successful query, xml response will have the IPAddress associated with the requested user-name and for failure query it returns 0.0.0.0 as the IPAddress.

## XML Interface

The XML interface is used for subscriber context information queries and responses to those queries. The XML interface is on a UDP port (8080) and is configurable. Identity caching supports the XML data-type definition (DTD) supported by the CSG.

The mapping from queries to replies can be one to many. For example, a UDP datagram might contain several queries but each reply will be returned in a separate datagram. No single query or reply can exceed the configured MTU of a datagram. Any that does results in an error.

If a query result is negative, the reply will consist of a null subscriber ID. All other error conditions cause Prime Access Registrar to drop the request. Errors are logged locally using the Prime Access Registrar logging mechanism.



## Using Prepaid Billing

---

Cisco Prime Access Registrar (Prime Access Registrar) supports two types of prepaid billing, IS835C and Cisco Real-time Billing (CRB), a Cisco proprietary solution. The IS835C version adheres to industry standards and is the preferred version.

Three components are required to support a prepaid billing service, such as the following:

- AAA client
- Prime Access Registrar server
- External prepaid billing server

The most important factor for an effective prepaid billing service is in developing a shared library to be configured under the prepaid RemoteServer object. The shared library should be developed to implement all specified API functions. You will have to provide a shared library that meets the needs of your environment. The shared library must implement the API functions to perform the various tasks required for your specific implementation of the prepaid billing service.



**Note**

---

Cisco works with you to develop the prepaid billing service and implement the API. For more information, contact your Cisco systems engineer.

---

The chapter contains the following sections:

- [Overview](#)
- [IS835C Prepaid Billing](#)
- [CRB Prepaid Billing](#)
- [Implementing the Prepaid Billing API](#)

# Overview

When a subscriber uses a prepaid billing service, each call requires a set of data about the subscriber. However, the AAA network has no previous knowledge of the subscriber's usage behavior. Prime Access Registrar uses an iterative authorization paradigm over multiple sessions to support the prepaid billing solution.

Each time an authorization request is made, the billing server apportions a fraction of the subscriber's balance into a quota. When a subscriber uses multiple sessions, each session must obtain its own quota. When a previously allocated quota is depleted, a session must be reauthorized to obtain a new quota.

**Note**

The granularity and the magnitude of the quota is in the design and implementation of the prepaid billing server and is beyond the scope of this document. In general, a smaller quota generates more network traffic, but allows more sessions per subscriber. When the quota is equal to a subscriber's total account balance, there is minimal network traffic, but only one session can be supported.

When a subscriber's current quota is depleted, the AAA client initiates a reauthorization request sending Access-Request packets. After the Prime Access Registrar server receives the request, it forwards the request to the billing server. The billing server then returns the next quota to use. The new quota might not be the same as the previous, and the billing server might adjust the quota dynamically.

## IS835C Prepaid Billing

Prime Access Registrar acts as a RADIUS protocol head for all the requirements specified in the *cdma2000 Wireless IP Network Standard: PrePaid Packet Data Service* specification:

[http://www.3gpp2.org/Public\\_html/specs/X.S0011-006-C-v1.0.pdf](http://www.3gpp2.org/Public_html/specs/X.S0011-006-C-v1.0.pdf)

As long as the prepaid client understands or accepts what the external billing server sends, the service should work. The Prime Access Registrar server neither imposes nor is affected by the values of attributes returned from the external billing server.

For additional information, see *cdma2000 Wireless IP Network Standard: Accounting Services and 3GPP2 RADIUS VSAs* at the following URL:

[http://www.3gpp2.org/Public\\_html/specs/X.S0011-005-C-v1.0.pdf](http://www.3gpp2.org/Public_html/specs/X.S0011-005-C-v1.0.pdf)

The IS835C specification requires that the Prime Access Registrar server be able to determine that a particular user is a prepaid billing user. A user is accepted as a valid prepaid user when the response dictionary of the incoming packet contains the Prime Access Registrar internal subattribute named *prepaid*.

The IS835C specification requires prepaid users to first be authenticated by the RADIUS server. This requires the configuration of a group service with an authentication service first, followed by the prepaid service that adds prepaid attributes as shown in [Setting Up an Authentication Group Service, page 8-5](#). The group service configuration enables the AA service to add the prepaid subattribute to the response dictionary upon successful authentication, before the prepaid service is invoked.

## Configuring IS835C Prepaid Billing

To configure an IS835C prepaid billing service, use the following sections to configure the required Prime Access Registrar objects:

- [Setting Up a Prepaid Billing RemoteServer](#)
- [Setting Up an IS835C Prepaid Service](#)
- [Setting Up Local Authentication](#)
- [Setting Up an Authentication Group Service](#)

### Setting Up a Prepaid Billing RemoteServer

Prime Access Registrar loads the library dynamically and registers the API functions, then calls out the library initialization API once at startup. The call to initialize functions initializes various data structures and connections with the billing server, as required.

[Table 8-1](#) lists and describes the properties required for an IS835C RemoteServer object.

**Table 8-1** Prepaid-IS835C RemoteServer Properties

Property	Description
Filename	Name of the shared library provided by the billing server vendor, such as <b>libprepaid.so</b>
IPAddress	IP address of the billing server
Port	Port used on the billing server, such as port 66
Connections	Number of threads the prepaid service and billing server can each use (default is 8).

#### Setting Up a Prepaid Billing Remote Server

To set up a prepaid billing remote server:

**Step 1** Use `aregcmd` to add a RemoteServer under `/Radius/RemoteServers`.

```
cd /radius/remoteserver
add prepaid-is835c
```

**Step 2** Set remoteserver protocol to prepaid-is835c.

```
cd prepaid-is835c
set protocol prepaid-is835c

Set Protocol prepaid-is835c
```

The following is the default configuration of a prepaid-is835c RemoteServer.

```
[ //localhost/Radius/RemoteServers/prepaid-is835c ]
Name = prepaid-is835c
Description =
Protocol =
IPAddress =
```

```
Port = 0
Filename =
Connections = 8
```

---

## Setting Up an IS835C Prepaid Service

Prime Access Registrar uses a service type **prepaid** to support the prepaid billing solution. The prepaid service mediates between the client NAS and the external prepaid billing server.

### Setting Up an IS835C Prepaid Service

To set up an IS835C prepaid service:

---

**Step 1** Use **aregcmd** to add a prepaid service under **/Radius/Services**:

```
cd /radius/services

add prepaid

Added prepaid
```

**Step 2** Set the service type to prepaid.

```
cd prepaid

set type prepaid

Set Type prepaid
```

A prepaid service has the following default properties:

```
[ //localhost/Radius/Services/prepaid ]
Name = prepaid
Description =
Type = prepaid
IncomingScript~ =
OutgoingScript~ =
OutagePolicy~ = RejectAll
OutageScript~ =
MultipleServersPolicy = Failover
RemoteServers/
```

**Step 3** Add a reference to the is835c-prepaid RemoteServer.

```
cd RemoteServer

add 1 prepaid-is835c

Added 1
```

---

## Setting Up Local Authentication

If you use the Prime Access Registrar server for authentication and authorization in your prepaid billing solution, you should configure an AA service. For example, you might configure a service similar to **local-users** (in the example configuration) for authentication and authorization of local users.

If some of the users are non-prepaid users or if the prepaid users need to have RADIUS authorization attributes returned, you should configure an AA service to perform that authentication and authorization.

### Setting Up a Local Authentication

To set up a local authentication:

**Step 1** Use **aregcmd** to set up a local authentication service.

```
cd /radius/services

add Prepaid-LocalAuthentication

Added prepaid-LocalAuthentication

cd prepaid-LocalAuthentication

[ //localhost/Radius/Services/prepaid-LocalAuthentication ]
  Name = prepaid-LocalAuthentication
  Description =
  Type =
```

**Step 2** Set the service type to local.

```
set type local

Set Type local
```

**Step 3** Set the **UserList** property to the userlist that contains IS835C prepaid users.

```
set UserList userlist_name

Set UserList userlist_name
```



**Note** You can use an LDAP or ODBC service in place of the local authentication service.

The authentication service must add the Prime Access Registrar internal attribute *prepaid* (subattribute 22) to the response upon successful authentication.

## Setting Up an Authentication Group Service

Your prepaid billing solution usually requires a group service to tie together an AA service with a prepaid service, a group service to tie together an accounting service with a prepaid service, or both.

If you are using an AA service with your prepaid billing solution, you must configure a group service, for example **prepaid-users**, that ties the requests to the AA service (**local-users** in our example) with the prepaid service.

If you are using Prime Access Registrar for an accounting service with your prepaid billing solution, you must configure a group service, for example **prepaid-file**, that ties accounting requests to both the regular accounting service (**local-file** in our example) and the prepaid service.

### Setting Up an Authentication Group Service

To set up an authentication group service:

---

**Step 1** Use **aregcmd** to add a prepaid authentication group service under **/Radius/Services**.

```
cd /radius/services
```

```
add prepaid-groupAuthentication
```

```
Added prepaid-groupAuthentication
```

```
cd prepaid-groupAuthentication
```

```
[ //localhost/Radius/Services/prepaid-groupAuthentication ]
  Name = group-prepaidAuthentication
  Description =
  Type =
```

**Step 2** Set the service type to group.

```
set type group
```

```
Set Type group
```

The group service requires the ResultRule to be set to AND, the default setting for a group service.

```
ls
```

```
[ //localhost/Radius/Services/group-prepaidAuthentication ]
  Name = group-prepaidAuthentication
  Description =
  Type = group
  IncomingScript~ =
  OutgoingScript~ =
  ResultRule = AND
  GroupServices/
```

**Step 3** Change directory to GroupServices and add references to the prepaid service and the authentication service.

```
cd GroupServices
```

```
[ //localhost/Radius/Services/group-prepaidAuthentication/GroupServices ]
```

```
add 1 Prepaid-LocalAuthentication
```

```
Added 1
```

```
add 2 prepaid
```

```
Added 2
```

---

# CRB Prepaid Billing

Cisco Real-Time Billing (CRB) is a Cisco proprietary method of providing prepaid billing service. Prime Access Registrar uses vendor-specific attributes (VSA) to extend the standard RADIUS protocol to carry information not usually present in the standard RADIUS packet. Prime Access Registrar uses a set of VSAs allocated to the Cisco VSA pool [26,9].

Prime Access Registrar required several different types of measurements to support a prepaid billing solution. These measurements require the use of metering variables to perform usage accounting. [Table 8-2](#) lists the different measurements and what the AAA client, Prime Access Registrar server, and billing server do with them.

**Table 8-2**      **Measurements and Component Actions**

Measurement Type	Billing Server Action	AAA Server Action	AAA Client Action
Duration	Return duration quota	Convert duration quota to VSAs and pass along	Compare running duration quota with quota returned by Prime Access Registrar server
Total volume	Return volume quota	Convert volume quota to VSAs and pass along	Compare running volume quota with quota returned by Prime Access Registrar server
Uplink volume	Return volume quota	Convert volume quota to VSAs and pass along	Compare running volume quota with quota returned by Prime Access Registrar server
Downlink volume	Return volume quota	Convert volume quota to VSAs and pass along	Compare running volume quota with quota returned by Prime Access Registrar server
Total packets	Return packet quota	Convert packet quota to VSAs and pass along	Compare running packet quota with quota returned by Prime Access Registrar server
Uplink packets	Return packet quota	Convert packet quota to VSAs and pass along	Compare running packet quota with quota returned by Prime Access Registrar server

**Table 8-2** *Measurements and Component Actions (continued)*

Measurement Type	Billing Server Action	AAA Server Action	AAA Client Action
Downlink packets	Return packet quota	Convert packet quota to VSAs and pass along	Compare running packet quota with quota returned by Prime Access Registrar server
Logical OR of two measurements	Return quota of both measurements	Convert both to VSA and pass along	Monitor both quota and issue reauthorization packet when any one trips

Prime Access Registrar provides maximum flexibility to billing servers by allowing the metering variable to be modified as the service is used. This requires network nodes to measure all parameters all the time, but to report values only after receiving a reauthorization request.

**Note**

If you have been using an earlier implementation of CRB prepaid billing (Cisco Access Registrar 3.5.2 or earlier), you must recompile the API implementation with the newer API due to the addition of the parameter `ebs_context` as the first parameter to all API methods. Contact your Cisco systems engineer for assistance with the new API.

This section contains the following topics:

- [Configuring CRB Prepaid Billing](#)
- [Configuring CRB Prepaid Billing for SSG](#)
- [Generic Call Flow](#)
- [Vendor-Specific Attributes](#)

## Configuring CRB Prepaid Billing

To configure an CRB prepaid billing service, use the following sections to configure the required Prime Access Registrar objects:

- [Setting Up a Prepaid Billing RemoteServer](#)
- [Setting Up a CRB Prepaid Service](#)
- [Setting Up a Local Accounting Service](#)
- [Setting Up a Local Authentication Service](#)
- [Setting Up a Prepaid Accounting Group Service](#)
- [Setting Up an Authentication Group Service](#)

If you are using CRB prepaid billing with Service Selection Gateway (SSG), you must also configure extension point scripts and prepaid clients. See [Configuring CRB Prepaid Billing for SSG, page 8-15](#).

### Setting Up a Prepaid Billing RemoteServer

[Table 8-3](#) lists and describes the properties required for an CRB RemoteServer object.

**Table 8-3** Prepaid-CRB RemoteServer Properties

Property	Description
Filename	Name of the shared library provided by the billing server vendor, such as <b>libprepaid.so</b>
IPAddress	IP address of the billing server
Port	Port used on the billing server, such as port 66
Connections	Number of threads the prepaid service and billing server can each use (default is 8).

**Setting Up a Prepaid Billing Remote Server**

To set up a prepaid billing remote server:

**Step 1** Use `aregcmd` to add a RemoteServer under **/Radius/RemoteServers**.

```
cd /radius/remoteservers
add prepaid-crb
Added prepaid-crb
```

**Step 2** Set the RemoteServer protocol to prepaid-crb.

```
cd prepaid-crb
set protocol prepaid-crb
Set Protocol prepaid-crb
```

The following is the default configuration of a prepaid-crb RemoteServer.

```
[ //localhost/Radius/RemoteServers/prepaid-crb ]
Name = prepaid-crb
Description =
Protocol =
IPAddress =
Port = 0
Filename =
Connections = 8
```

**Setting Up a CRB Prepaid Service**

Prime Access Registrar uses a service type **prepaid** to support the prepaid billing solution. The prepaid service mediates between the client NAS and the external prepaid billing server.

The prepaid service must receive accounting requests to accurately charge the prepaid billing user. You can also set the prepaid service in a group service to log accounting requests locally or to proxy the accounting requests to another service or to both locations.

### Setting Up a CRB Prepaid Service

To set up a CRB prepaid service:

---

**Step 1** Use `aregcmd` to add a prepaid service under `/Radius/Services`:

```
cd /radius/services
```

```
add prepaid
```

```
Added prepaid
```

**Step 2** Set the service type to prepaid.

```
cd prepaid
```

```
set type prepaid
```

```
Set Type prepaid
```

A prepaid service has the following default properties:

```
[ //localhost/Radius/Services/prepaid ]
Name = prepaid
Description =
Type = prepaid
IncomingScript~ =
OutgoingScript~ =
OutagePolicy~ = RejectAll
OutageScript~ =
MultipleServersPolicy = Failover
RemoteServers/
```

**Step 3** Add a reference to the prepaid-crb RemoteServer.

```
cd RemoteServers
```

```
add 1 prepaid-crb
```

```
Added 1
```




---

**Note** The following steps are required only when using Prepaid-CRB with SSG.

---

**Step 4** Set the IncomingScript to **IncomingScript PPI-Parse-Prepaid-Incoming**.

```
set IncomingScript PPI-Parse-Prepaid-Incoming
```

```
Set IncomingScript PPI-Parse-Prepaid-Incoming
```

**Step 5** Set the OutgoingScript to **PPO-Parse-Prepaid-Outgoing**.

```
set OutgoingScript PPO-Parse-Prepaid-Outgoing
```

```
Set OutgoingScript PPO-Parse-Prepaid-Outgoing
```

---

## Setting Up a Local Accounting Service

If you want to use the Prime Access Registrar server to record the accounting records locally or to forward the accounting records to another RADIUS server, you must configure an accounting service. You might configure a service similar to **local-file** (in the example configuration) for accounting requests. Accounting requests can be logged locally (with an accounting service) or remotely (with a RADIUS service).

If you use the prepaid billing server to generate the accounting records, an accounting service is not necessary.

### Setting Up a Local Accounting Service

To set up a local accounting service:

**Step 1** Use **aregcmd** to add a local accounting service under **/Radius/Services**.

```
cd /radius/services

add prepaid-LocalFileAccounting

add prepaid-LocalFileAccounting
```

**Step 2** Set the service type to file.

```
cd prepaid-LocalFileAccounting

set type file

Set Type file
```

The file type service has the following properties:

```
[ //localhost/Radius/Services/prepaid-LocalFileAccounting ]
Name = prepaid-LocalFileAccounting
Description =
Type = file
IncomingScript~ =
OutgoingScript~ =
OutagePolicy~ = RejectAll
OutageScript~ =
FilenamePrefix = accounting
MaxFileSize = "10 Megabytes"
MaxFileAge = "1 Day"
RolloverSchedule =
UseLocalTimeZone = FALSE
```

**Step 3** Set the **FilenamePrefix** to **Prepaid-Accounting**.

```
set FilenamePrefix Prepaid-Accounting

Set FilenamePrefix Prepaid-Accounting
```

**Step 4** Set the **MaxFileAge** to one hour.

```
set MaxFileAge "1 Hour"

Set MaxFileAge "1 Hour"
```

The **MaxFileSize** should remain at the default value of 10 megabytes.

**Step 5** Set `UseLocalTimeZone` to `TRUE`.

```
set UseLocalTimeZone TRUE
```

```
Set UseLocalTimeZone TRUE
```

---

## Setting Up a Local Authentication Service

If you use the Prime Access Registrar server for authentication and authorization in your prepaid billing solution, you should configure an AA service. For example, you might configure a service similar to **local-users** (in the example configuration) for authentication and authorization of local users.

If some of the users are non-prepaid users or if the prepaid users need to have RADIUS authorization attributes returned, you should configure an AA service to perform that authentication and authorization.

If all of the users in a realm are prepaid users and the prepaid billing client does not require normal RADIUS authorization attributes, an AA service is not necessary.

### Setting Up a Local Authentication Service

To set up a local authentication service:

---

**Step 1** Use `aregcmd` to set up a local authentication service.

```
cd /radius/services
```

```
add Prepaid-LocalAuthentication
```

```
Added prepaid-LocalAuthentication
```

```
cd prepaid-LocalAuthentication
```

```
[ //localhost/Radius/Services/prepaid-LocalAuthentication ]
  Name = prepaid-LocalAuthentication
  Description =
  Type =
```

**Step 2** Set the service type to local.

```
set type local
```

```
Set Type local
```

**Step 3** Set the `UserList` property to the userlist that contains IS835C prepaid users.

```
set UserList userlist_name
```

```
Set UserList userlist_name
```



#### Note

You can use an LDAP or ODBC service in place of the local authentication service.

---

## Setting Up a Prepaid Accounting Group Service

A prepaid billing solution usually requires a group service to tie together an AA service with a prepaid service, a group service to tie together an accounting service with a prepaid service, or both.

If you are using an AA service with your prepaid billing solution, you must configure a group service, for example **prepaid-users**, that ties the requests to the AA service (**local-users** in our example) with the prepaid service.

If you are using Prime Access Registrar for an accounting service with your prepaid billing solution, you must configure a group service, for example **prepaid-file**, that ties accounting requests to both the regular accounting service (**local-file** in our example) and the prepaid service.

### Setting Up a Prepaid Accounting Group Service

To set up a prepaid accounting group service:

**Step 1** Use `aregcmd` to create an accounting group service under `/Radius/Services`.

```
cd /radius/services

add Prepaid-Accounting

Added prepaid-accounting
```

**Step 2** Set the service type to group.

```
cd prepaid-accounting

[ //localhost/Radius/Services/prepaid-accounting ]
Name = prepaid-accounting
Description =
Type =

set type group

Set Type group
```

The group service has the following properties:

```
[ //localhost/Radius/Services/prepaid-accounting ]
Name = prepaid-accounting
Description =
Type = group
IncomingScript~ =
OutgoingScript~ =
ResultRule = AND
GroupServices/
```

**Step 3** Reference the Prepaid and Prepaid-LocalAccounting services under GroupServices.

```
cd GroupServices

[ //localhost/Radius/Services/prepaid-accounting/GroupServices ]

add 1 prepaid

Added 1
```

**add 2 prepaid-LocalFileAccounting**

```
Added 2
```

---

**Setting Up an Authentication Group Service**

A prepaid billing solution usually requires a group service to tie together an AA service with a prepaid service, a group service to tie together an accounting service with a prepaid service, or both.

If you are using an AA service with your prepaid billing solution, you must configure a group service, for example **prepaid-users**, that ties the requests to the AA service with the prepaid service.

If you are using Prime Access Registrar for an accounting service with your prepaid billing solution, you must configure a group service, for example **prepaid-file**, that ties accounting requests to both the regular accounting service and the prepaid service.

**Setting Up an Authentication Group Service**

To set up an authentication group service:

- 
- Step 1** Use **aregcmd** to add a prepaid authentication group service under **/Radius/Services**.

```
cd /radius/services
```

```
add prepaid-groupAuthentication
```

```
Added group-prepaidAuthentication
```

```
cd group-prepaidAuthentication
```

```
[ //localhost/Radius/Services/group-prepaidAuthentication ]
  Name = group-prepaidAuthentication
  Description =
  Type =
```

- Step 2** Set the service type to group.

```
set type group
```

```
Set Type group
```

The group service requires the ResultRule to be set to AND, the default setting for a group service.

```
ls
```

```
[ //localhost/Radius/Services/group-prepaidAuthentication ]
  Name = group-prepaidAuthentication
  Description =
  Type = group
  IncomingScript~ =
  OutgoingScript~ =
  ResultRule = AND
  GroupServices/
```

- Step 3** Change directory to GroupServices and add references to the prepaid service and the authentication service.

**cd GroupServices**

```
[ //localhost/Radius/Services/group-prepaidAuthentication/GroupServices ]
```

**add 1 Prepaid-LocalAuthentication**

```
Added 1
```

**add 2 prepaid**

```
Added 2
```

---

## Configuring CRB Prepaid Billing for SSG

In addition to the configuration described in [CRB Prepaid Billing, page 8-7](#), when using CRB-Prepaid billing with SSG, you must also perform the following:

- [Setting Up an Outgoing Script](#)
- [Setting Up an Incoming Script](#)
- [Setting Up a Prepaid Outgoing Script](#)
- [Adding Prepaid Clients](#)

### Setting Up an Outgoing Script

To set up an outgoing script:

- Step 1** Use **aregcmd** to add the **PCO-Parse-Client-Outgoing** outgoing script under **/Radius/Scripts**:

**cd /radius/scripts**

**add PCO-Parse-Client-Outgoing**

```
Added PCO-Parse-Client-Outgoing
```

**cd PCO-Parse-Client-Outgoing**

```
[ //localhost/Radius/Scripts/PCO-Parse-Client-Outgoing ]
Name = PCO-Parse-Client-Outgoing
Description =
Language =
```

- Step 2** Set the language to tcl.

**set language tcl**

```
Set Language tcl
```

- Step 3** Set the filename to **PCO-parse.client-outgoing.tcl**.

**set filename PCO-parse.client-outgoing.tcl**

```
Set Filename PCO-parse-client-outgoing.tcl
```

**Step 4** Set the EntryPoint to PCO-parse-client-outgoing.

```
set EntryPoint PCO-parse-client-outgoing
```

```
Set EntryPoint PCO-parse-client-outgoing
```

---

### Setting Up an Incoming Script

To set up an incoming script:

---

**Step 1** Use **aregcmd** to add the **PPI-Parse-Prepaid-Incoming** script under **/Radius/Scripts**.

```
cd /radius/scripts
```

```
add PPI-Parse-Prepaid-Incoming
```

**Step 2** Set the language to tcl.

```
cd PPI-Parse-Prepaid-Incoming
```

```
set language tcl
```

```
Set Language tcl
```

**Step 3** Set the filename to **PPI-Parse-Prepaid-Incoming.tcl**.

```
set filename PPI-Parse-Prepaid-Incoming.tcl
```

```
Set Filename PPI-Parse-Prepaid-Incoming.tcl
```

**Step 4** Set the EntryPoint to PPO-Parse-Prepaid-Outgoing.

```
set EntryPoint PPO-Parse-Prepaid-Outgoing
```

```
Set EntryPoint PPO-Parse-Prepaid-Outgoing
```

---

### Setting Up a Prepaid Outgoing Script

To set up a prepaid outgoing script:

---

**Step 1** Use **aregcmd** to add the **PPO-Parse-Prepaid-Outgoing** outgoing script under **/Radius/Scripts**:

```
cd /radius/scripts
```

**Step 2** Add the **PPO-Parse-Prepaid-Outgoing** outgoing script under **/Radius/Scripts**.

```
cd /radius/scripts
```

```
add PPO-Parse-Prepaid-Outgoing
```

```
Added PPO-Parse-Prepaid-Outgoing
```

- Step 3** Set the language to tcl.
- ```
cd PPO-Parse-Prepaid-Outgoing
set language tcl
Set Language tcl
```
- Step 4** Set the filename to **PPO-Parse-Prepaid-Outgoing.tcl**.
- ```
set filename PPO-Parse-Prepaid-Outgoing.tcl
Set Filename PPO-Parse-Prepaid-Outgoing.tcl
```
- Step 5** Set the EntryPoint to PPO-Parse-Prepaid-Outgoing.
- ```
set EntryPoint PPO-Parse-Prepaid-Outgoing
Set EntryPoint PPO-Parse-Prepaid-Outgoing
```
- 

### Adding Prepaid Clients

To add prepaid clients:

- Step 1** Use **aregcmd** to add the prepaid clients under **/Radius/Clients**.

```
cd /radius/clients
add SSG
```

A RADIUS client has the following properties:

```
[ //localhost/Radius/Clients/ssg ]
Name = ssg
Description =
IPAddress =
SharedSecret =
Type = NAS
Vendor =
IncomingScript~ =
OutgoingScript~ =
EnableDynamicAuthorization = FALSE
NetMask =
```

- Step 2** Set the **IPAddress** property to the client IP address.

```
set IPAddress aaa.bbb.ccc.ddd
Set IPAddress aaa.bbb.ccc.ddd
```

- Step 3** Set the **SharedSecret**.

```
set sharedsecret cisco
Set SharedSecret cisco
```

- Step 4** Set the to **PCO-Parse-Client-Outgoing**.

**set out PCO-Parse-Client-Outgoing**

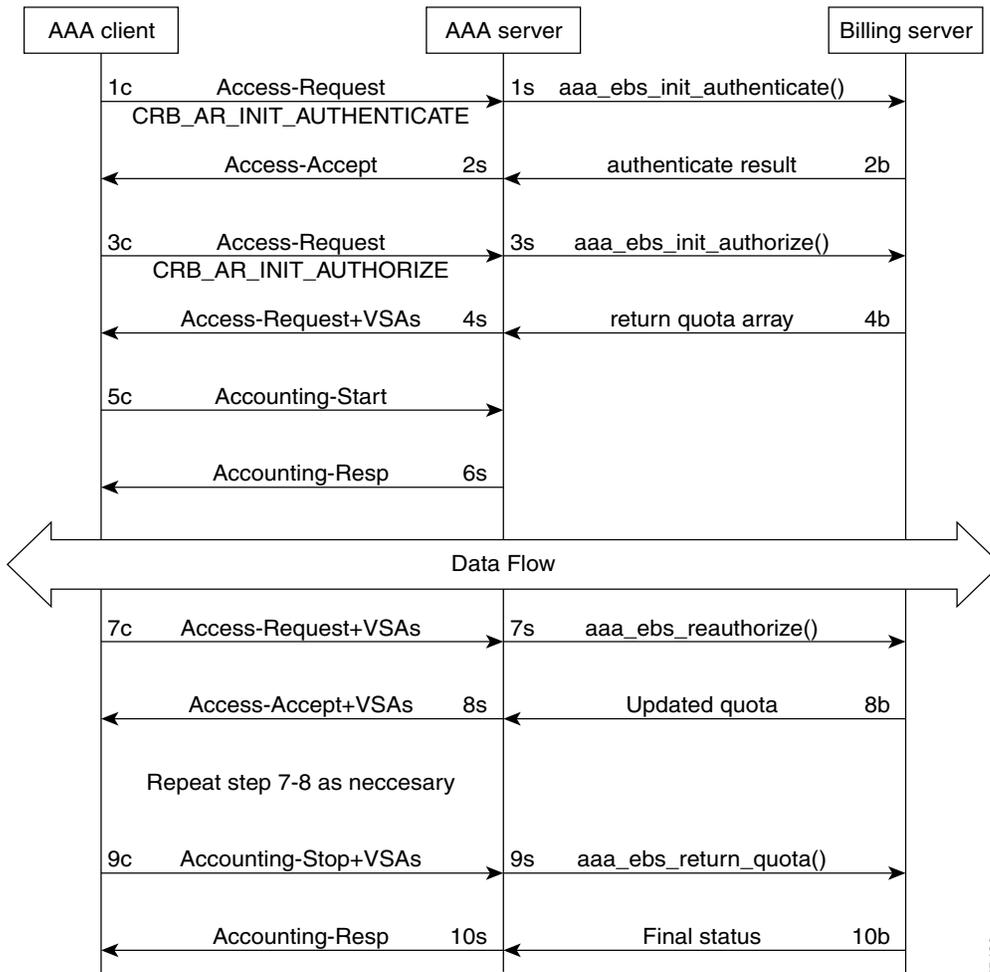
Set PCO-Parse-Client-Outgoing

## Generic Call Flow

This section describes the generic call flow for the Prime Access Registrar CRB prepaid billing. The call flow is controlled by the AAA client. The Prime Access Registrar server converts VSAs into calls to the billing server. For information about call flows and attributes for IS835C, see [IS835C Prepaid Billing, page 8-2](#).

The packet flows presented in [Figure 8-1](#) are specific to the Prime Access Registrar CRB prepaid billing only. The headlines in the packet flows are general and do represent all data transferred. The letters **c**, **s**, and **b** in [Figure 8-1](#) designate the packet's source of **client**, **server**, or **billing server**, respectively.

**Figure 8-1 Generic Call Flow Diagram**



75496

This section contains the following topics:

- [Access-Request \(Authentication\)](#)
- [Access-Accept \(Authentication\)](#)
- [Access-Request \(Authorization\)](#)
- [Access-Accept \(Authorization\)](#)
- [Accounting-Start](#)
- [Data Flow](#)
- [Access-Request \(Quota Depleted\)](#)
- [Accept-Accept \(Quota Depleted\)](#)
- [Accounting Stop \(Session End\)](#)
- [Accounting Response \(Final Status\)](#)

## Access-Request (Authentication)

**Flow 1c** shows the client sending the Access-Request to AAA Server, part of a normal authentication request. The exact nature of the message contents is dictated by the access technology, be it be CDMA1X-RTT, GPRS, or another. The Access-Request might involve other messages such as PAP/CHAP or another form of authentication.

The **Flow 1c** Access-Request might contain a prepaid specific VSA, CRB\_AUTH\_REASON. [Table 8-4](#) lists the attributes included in the authentication Access-Request. This tells the Prime Access Registrar server to authenticate the subscriber with the Prepaid server as well. If the value is CRB\_AR\_INIT\_AUTHENTICATE, the initial quota must be obtained for a single service prepaid solution. If this VSA is not present, the Prime Access Registrar server will not authenticate with the Prepaid billing server.

**Table 8-4** *Attributes Sent During Subscriber Authentication*

| Attribute Number | Attribute Name                              | Description                      | Notes                               |
|------------------|---|----------------------------------|-------------------------------------|
| 1                | User-Name                                   | APPL: Mobile Node Username       | Required                            |
| 2                | NAS IP Address                              | Accounting Node IP Address       | APPL: Required, POA                 |
| 31               | Calling-station-ID                          | APPL:MSISDN or IMSI              | APPL: Conditional                   |
| 26, 9            | CRB_AUTH_REASON<br>CRB_AR_INIT_AUTHENTICATE | See VSA section                  | Required                            |
| 26, 9            | CRB_USER_ID                                 | APPL:PDSN address or SSG address | APPL: Required, Address of the PDSN |

**Table 8-4** *Attributes Sent During Subscriber Authentication (continued)*

| Attribute Number | Attribute Name | Description   | Notes  |
|------------------|----------------|---|--|
| 26, 9            | CRB_SERVICE_ID | APPL: Service ID such as Simple IP service, Mobile IP service, or VPN service |  |
| 26, 9            | CRB_SESSION_ID | This VSA contains the session key ID information                              | Required; the session ID must be globally unique across all clients and across reboots of the client |

In **Flow 1s**, the Prime Access Registrar server sends a call to the billing server to authenticate the prepaid user and possibly determine more information about the subscriber's account. The Prime Access Registrar server can be configured to generate this packet flow, using a subscriber profile parameter, if the request is from a prepaid subscriber.

## Access-Accept (Authentication)

**Flow 2b** shows the billing server returning the authentication result. The billing server returns a failure if the prepaid subscriber has an inadequate balance.

**Flow 2s** shows the Prime Access Registrar server sending the Access-Accept to the AAA client. This message flow contains at least one prepaid billing-specific VSA (listed in [Table 8-5](#)) and might contain other access technology-specific attributes.

**Table 8-5** *Attributes Sent to AAA client in Access-Accept (Authentication)*

| Attribute Number | Attribute Name                             | Description   | Notes    |
|------------------|--|---|----------|
| 26, 9            | CRB__USER_TYPE<br>CRB_AR_INIT_AUTHENTICATE | See <a href="#">Vendor-Specific Attributes, page 8-25</a> | Optional |

## Access-Request (Authorization)

In **Flow 3c**, the AAA client sends another Access-Request, this time to authorize the subscriber. [Table 8-6](#) lists the attributes required by the Prime Access Registrar server to authorize the subscriber. The session key ID used must be specified using a prepaid VSA pointing to the RADIUS attribute (standard or VSA).

**Table 8-6** *Attributes Sent During Subscriber Authorization*

| Attribute Number | Attribute Name | Description                | Notes               |
|------------------|----------------|----------------------------|---------------------|
| 1                | User-Name      | APPL: Mobile Node Username | Required            |
| 2                | NAS IP Address | Accounting Node IP Address | APPL: Required, POA |

**Table 8-6** *Attributes Sent During Subscriber Authorization (continued)*

| Attribute Number | Attribute Name                           | Description   | Notes  |
|------------------|--|---|--|
| 31               | Calling-station-ID                       | APPL:MSISDN or IMSI   | APPL: Conditional  |
| 26, 9            | CRB_AUTH_REASON<br>CRB_AR_INIT_AUTHORIZE | See <a href="#">Vendor-Specific Attributes, page 8-25</a>                     | Required   |
| 26, 9            | CRB_USER_ID                              | APPL:PDSN address or SSG address  | APPL: Required, Address of the PDSN  |
| 26, 9            | CRB_SERVICE_ID                           | APPL: Service ID such as Simple IP service, Mobile IP service, or VPN service |  |
| 26, 9            | CRB_SESSION_ID                           | This VSA contains the session key ID information                              | Required; the session ID must be globally unique across all clients and across reboots of the client |

.In **Flow 3s**, the Prime Access Registrar server sends the Prepaid billing server to obtain a quota. The quota might contain several values depending on the number of measurement parameters chosen.

## Access-Accept (Authorization)

**Flow 4b** shows the billing server returning the quota array for the subscriber.

In **Flow 4s**, the Prime Access Registrar server converts the quota array received into VSAs and sends an Access-Accept with the assembled VSAs to the AAA client. [Table 8-7](#) lists the prepaid-specific VSAs that might be included in the Access-Accept response message sent to the AAA client. For more detailed information about the VSAs, see [Vendor-Specific Attributes, page 8-25](#).

**Table 8-7** *Attributes Sent to AAA client in Access-Accept (Authorization)*

| Attribute Number | Attribute Name       |
|------------------|----------------------|
| 26, 9            | CRB_DURATION         |
| 26, 9            | CRB_TOTAL_VOLUME     |
| 26, 9            | CRB_UPLINK_VOLUME    |
| 26, 9            | CRB_DOWNLINK_VOLUME  |
| 26, 9            | CRB_TOTAL_PACKETS    |
| 26, 9            | CRB_UPLINK_PACKETS   |
| 26, 9            | CRB_DOWNLINK_PACKETS |

**Flows 3c** through **4s** are repeated for every service started or restarted by the AAA client.

However, if the return parameters indicate that the authorization is rejected, an Access-Accept message is generated and sent to the client as shown in [Table 8-8](#). When this type of error condition occurs, no other VSA is included in the Access-Accept message.

**Table 8-8** Attribute Sent to Report Error Condition to AAA client

| Attribute Number | Attribute Name      | Description   | Notes  |
|------------------|---------------------|---|--|
| 26, 9            | CRB_TERMINATE_CAUSE | Identifies why a subscriber failed authentication:<br>1. Exceeded the balance<br>2. Exceeded the overdraft<br>3. Bad credit<br>4. Services suspended<br>5. Invalid User | Conditional; rejection might be returned with Access-Accept and zero (0) quota |

## Accounting-Start

In **Flow 5c**, the AAA client sends the Accounting-Start. In **Flow 6s**, the Prime Access Registrar server replies with the Accounting-Response.

## Data Flow

At this point, the data transfer begins. The AAA client monitors the subscriber's allocated quotas for metering parameters. A subscriber's Reauthorization request is generated when a quota for at least one of the metering parameters, is depleted.

## Access-Request (Quota Depleted)

**Flow 7c** shows the client sending an Access-Request to the Prime Access Registrar server because at least one quota has been depleted. The Access-Request includes different measurements of how much of the quotas were used in VSA format. This enables the billing server to account for the usage and manage the subscriber's balance before assigning a new quota. [Table 8-9](#) lists the attributes returned to the Prime Access Registrar server:

**Table 8-9** Attributes Sent by NAS When Quota Depleted

| Attribute Number | Attribute Name     | Description                       | Notes   |
|------------------|--------------------|-----------------------------------|---|
| 1                | User-Name          | APPL: Mobile Node Username        | Conditional                                       |
| 2                | NAS IP Address     | Accounting Node IP Address        | APPL: Required, POA address, or Home Node address |
| 31               | Calling-station-ID | APPL:MSISDN or IMSI               | APPL: Conditional                                 |
| 26, 9            | CRB_AUTH_REASON    | See VSA                           | Required  |
| 26, 9            | CRB_USER_ID        | APPL: PDSN address or SSG address | APPL: Required, address of SGSN                   |

**Table 8-9** *Attributes Sent by NAS When Quota Depleted (continued)*

| Attribute Number | Attribute Name       | Description   | Notes       |
|------------------|----------------------|---|-------------|
| 26, 9            | CRB_DURATION         | See <a href="#">Vendor-Specific Attributes, page 8-25</a> | Required    |
| 26, 9            | CRB_TOTAL_VOLUME     |   | Conditional |
| 26, 9            | CRB_UPLINK_VOLUME    |   |             |
| 26, 9            | CRB_DOWNLINK_VOLUME  |   |             |
| 26, 9            | CRB_TOTAL_PACKETS    |   |             |
| 26, 9            | CRB_UPLINK_PACKETS   |   |             |
| 26, 9            | CRB_DOWNLINK_PACKETS |   |             |

## Accept-Accept (Quota Depleted)

**Flow 7s** shows the Prime Access Registrar server returning the used quota array to the billing server. The call includes `aaa_ibs_reauthoriz()`. The billing server sends an updated quota array for the next period to the Prime Access Registrar server.

In **Flow 8s**, the Prime Access Registrar server converts the quota array into VSAs and sends them to the AAA client.

**Table 8-10** *Attributes Sent to AAA Client in Access-Accept (Reauthorization)*

| Attribute Number | Attribute Name       |
|------------------|----------------------|
| 26, 9            | CRB_USER_TYPE        |
| 26, 9            | CRB_DURATION         |
| 26, 9            | CRB_TOTAL_VOLUME     |
| 26, 9            | CRB_UPLINK_VOLUME    |
| 26, 9            | CRB_DOWNLINK_VOLUME  |
| 26, 9            | CRB_TOTAL_PACKETS    |
| 26, 9            | CRB_UPLINK_PACKETS   |
| 26, 9            | CRB_DOWNLINK_PACKETS |

## Accounting Stop (Session End)

In **Flow 9c**, the client sends an Accounting-Stop to the Prime Access Registrar server to end the session. The Accounting-Stop message includes an updated quota array with the usage adjustments since the previous authorization in the VSA form.

[Table 8-11](#) lists the attributes included in the Accounting-Stop message set to the Prime Access Registrar server and forwarded to the billing server.

## Accounting Response (Final Status)

In **Flow 9s**, the Prime Access Registrar server sends the used quota array to the billing server in an Accounting-Stop message. Any values returned by the billing server in **Flow 10b** are discarded.

**Flow 10s** shows the Prime Access Registrar server sending final Accounting-Response message to the AAA client.

**Table 8-11** *Attributes Sent in Accounting-Stop Message*

| Attribute Number | Attribute Name       | Description  | Notes   |
|------------------|----------------------|--|---|
| 1                | User-Name            | APPL: Mobile Node Username   | Conditional   |
| 2                | NAS IP Address       | Accounting Node IP Address   | APPL: Required, POA   |
| 31               | Calling-station-ID   | APPL:MSISDN or IMSI  | APPL: Conditional   |
| 40, 2            | Acct_status_type     | Indicates the accounting “Stop” for the service  | Required; this value (2) indicates an Accounting-Stop request message |
| 42               | Acct-Input-Octets    | The number of octets sent by the subscriber; uplink  | Required  |
| 43               | Acc_Output_Octets    | The number of octets received by the subscriber; downlink  |   |
| 46               | Acct-Session-Time    | Duration of the session  |   |
| 47               | Acct-Input-Packets   | Number of packets sent by the subscriber   |   |
| 48               | Acct-Output-Packets  | Number of packets received by the subscriber   |   |
| 49               | Acct-Terminate-Cause | This parameter, used for tracking, should remain the same for all accounting requests for a given service. |   |
| 26, 9            | CRB_DURATION         | See <a href="#">Vendor-Specific Attributes, page 8-25</a>  | Conditional   |
| 26, 9            | CRB_TOTAL_VOLUME     |  |   |
| 26, 9            | CRB_UPLINK_VOLUME    |  |   |
| 26, 9            | CRB_DOWNLINK_VOLUME  |  |   |
| 26, 9            | CRB_TOTAL_PACKETS    |  |   |
| 26, 9            | CRB_UPLINK_PACKETS   |  |   |
| 26, 9            | CRB_DOWNLINK_PACKETS |  |   |
| 26, 9            | CRB_SESSION_ID       | Specifies the RADIUS attribute carrying the session ID information   | Optional  |

## Vendor-Specific Attributes

Vendor-specific attributes are included in specific RADIUS packets to communicate prepaid user balance information from the Prime Access Registrar server to the AAA client, and actual usage, either interim or total, between the NAS and the Prime Access Registrar Server.

Table 8-12 lists the VSAs that will be defined in the API. Table 8-12 also lists the string to be used with Cisco-AVPair below the VSA.


**Note**

VSAs that start with CRB are used for Cisco Radius Billing prepaid service.

**Table 8-12 Vendor-Specific Attributes for the Cisco Prepaid Billing Solution**

| VSA Name                           | Type   | Source (Call Flow) | Description   |
|------------------------------------|--------|--------------------|---|
| CRB_AUTH_REASON<br>crb-auth-reason | Int8   | 1c, 7c, 7'c        | Passed with re-authorization:<br>1. Initial Authentication<br>2. Initial Authorization<br>3. Re-authorization<br>4. Return Quota<br>5. Query to EBS   |
| CRB_USER_ID<br>crb-user-id         | String | 1c, 7c, 7'c        | APPL: In PDSN this can be Address of the PDSN.  |
| CRB_SERVICE_ID<br>crb-service-id   | String | 1c, 7c             | Identifies the subscriber's service   |
| CRB_USER_TYPE<br>crb-entity-type   | Int8   | 4s                 | Type of user:<br>1. Prepaid user<br>2. Post-paid with no credit limit<br>3. Post-paid with credit limit<br>4. Invalid user<br><br>The source for this VSA value could be from the Subscriber profile or from the billing server |

**Table 8-12 Vendor-Specific Attributes for the Cisco Prepaid Billing Solution (continued)**

| <b>VSA Name</b>                              | <b>Type</b> | <b>Source (Call Flow)</b> | <b>Description</b>                               |
|--|-------------|---------------------------|--|
| CRB_DURATION<br>crb-duration                 | Int32       | 4s, 8s                    | Downlink quota received by the AAA client        |
| CRB_TOTAL_VOLUME<br>crb-total-volume         |             |                           | Total Volume quota received by the AAA client    |
| CRB_UPLINK_VOLUME<br>crb-uplink-volume       |             |                           | Uplink volume quota received by the AAA client   |
| CRB_DOWNLINK_VOLUME<br>crb-downlink-volume   |             |                           | Uplink Volume quota received by the AAA client   |
| CRB_TOTAL_PACKETS<br>crb-total-packets       |             |                           | Downlink Packet quota received by the AAA client |
| CRB_UPLINK_PACKETS<br>crb-uplink-packets     |             |                           | Uplink Packet quota received by the AAA client   |
| CRB_DOWNLINK_PACKETS<br>crb-downlink-packets |             |                           | Uplink Volume quota received by the AAA client   |
| CRB_SESSION_ID<br>crb-session-id             |             |                           | String   |

**Table 8-12 Vendor-Specific Attributes for the Cisco Prepaid Billing Solution (continued)**

| VSA Name                                   | Type   | Source (Call Flow) | Description   |
|--|--------|--------------------|---|
| CRB_TERMINATE_CAUSE<br>crb-terminate-cause | Int8   | 4se                | Identifies why a subscriber failed authentication:<br><ol style="list-style-type: none"> <li>1. Exceeded the balance</li> <li>2. Exceeded the overdraft</li> <li>3. Bad credit</li> <li>4. Services suspended</li> <li>5. Invalid User</li> <li>6. Invalid Password</li> <li>7. System Error</li> <li>8. Disabled</li> <li>9. Expired</li> <li>10. Valid in Future</li> <li>11. Used up</li> <li>12. No Parallel sessions</li> <li>13. Session Already closed</li> <li>14. Invalid session</li> </ol> |
| CRB_PRIVATE<br>crb-private                 | String | n/a                | Reserved for future use   |

## Implementing the Prepaid Billing API

A shared library must implement the API functions to perform the various tasks given in the description of each of the function. This needs to be compiled as a shared library and then specified as part of the remote server configuration at the Filename property. See [Setting Up a Prepaid Billing RemoteServer, page 8-3](#) or [Setting Up a Prepaid Billing RemoteServer, page 8-8](#).

At startup, Prime Access Registrar loads the library dynamically and registers the API functions, then calls out the library initialization API once at startup. The call to initialize functions initializes various data structures and connections with the billing server, as required.



### Note

Cisco works with you to develop the prepaid billing service and implement the API. For more information, contact your Cisco systems engineer.

At various times, according to the call flow described in the Prepaid Call Flow Specification (CRB or IS835C), Prime Access Registrar calls out appropriate API functions present in the shared library. The values for the arguments passed to these API calls are purely derived from the incoming RADIUS packet and Prime Access Registrar does not maintain any dynamic information related to the call flow. It is up to the API function to make use of the information passed to it as C structures to contact the Billing server, get appropriate data, and return the same to Prime Access Registrar using the designated arguments.



### Note

See the API specifications for more details pertaining to the arguments and return values of the API.





# Using Cisco Prime Access Registrar Server Features

---

This chapter provides information about how to use the Cisco Prime Access Registrar (Prime Access Registrar) server features.

This chapter contains the following sections:

- [Incoming Traffic Throttling](#)
- [Backing Store Parsing Tool](#)
- [Configurable Worker Threads Enhancement](#)
- [Session-Key Lookup](#)
- [Query-Notify](#)
- [Support for Windows Provisioning Service](#)
- [Command Completion](#)
- [Service Grouping Feature](#)
- [SHA-1 Support for LDAP-Based Authentication](#)
- [Dynamic Attributes](#)
- [Tunneling Support Feature](#)
- [xDSL VPI/VCI Support for Cisco 6400](#)
- [Apply Profile in Cisco Prime Access Registrar Database to Directory Users](#)
- [Directory Multi-Value Attributes Support](#)
- [MultiLink-PPP \(ML-PPP\)](#)
- [Dynamic Updates Feature](#)
- [NAS Monitor](#)
- [Automatic Information Collection \(arbug\)](#)
- [Simultaneous Terminals for Remote Demonstration](#)
- [Support for RADIUS Check Item Attributes](#)
- [User-Specific Attributes](#)
- [Packet of Disconnect](#)
- [Dynamic DNS](#)

- [Dynamic Service Authorization Feature](#)
- [Remote Session Management](#)
- [Wx Interface Support for SubscriberDB Lookup](#)
- [Smart Grid Solution Management](#)
- [Lawful Interception \(LI\) Support in Prime Access Registrar](#)
- [TACACS+ Support for AAA](#)

## Incoming Traffic Throttling

Prime Access Registrar offers two options to tackle traffic bursts by limiting incoming traffic. You will find two properties, `MaximumIncomingRequestRate` and `MaximumOutstandingRequests`, under `/Radius/Advanced` to limit the incoming traffic.

This contains the following sections:

- [MaximumIncomingRequestRate](#)
- [MaximumOutstandingRequests](#)

### MaximumIncomingRequestRate

You can use the `MaximumIncomingRequestRate` property to limit incoming traffic in terms of “allowed requests per second”.

For example, if you set the `MaximumIncomingRequestRate` to  $n$ , then at any given second, only  $n$  requests are accepted for processing. In the next second, another  $n$  requests are accepted regardless of whether the requests accepted earlier are processed or not. This condition serves as a soft limit.

The `MaximumIncomingRequestRate` property by default is zero (disabled).

### MaximumOutstandingRequests

You can use the `MaximumOutstandingRequests` property to limit incoming traffic in terms of “requests processed”.

For example, if you set the `MaximumOutstandingRequests` to  $n$ ,  $n$  requests are accepted for processing. Further requests are accepted only after processing some of these requests and sending the replies back. This condition serves as a hard limit.

The `MaximumOutstandingRequests` property by default is zero (disabled).



#### Note

You can enable either of these properties independent of the other.

#### Configuring the MaximumOutstandingRequests

To configure the `MaximumIncomingRequestRate` or `MaximumOutstandingRequests` property:

- Step 1** Log into `aregcmd`.
- Step 2** Change directory to `/Radius/Advanced`.

**Step 3** Set the `MaximumIncomingRequestRate` or `MaximumOutstandingRequests` property to non-zero values.

```
set MaximumIncomingRequestRate n
```

or

```
set MaximumOutstandingRequests n
```

where *n* is any nonzero value.

**Step 4** Save the configuration; enter:

```
save
```

**Step 5** Reload the server; enter:

```
reload
```

---

## Backing Store Parsing Tool

Prime Access Registrar tool, **carbs.pl**, helps to analyze the session backing store files. You will find this tool under `/cisco-ar/bin` directory.

Using `carbs.pl`, you can:

- Get information about the active, stopped, and stale RADIUS sessions.
- Clear phantom sessions manually.
- Process the binary log files and get information in a user-readable format.

The syntax is:

```
carbs.pl [-a] [-d <dir>] [-f <logfile>] [-v] [p] [-o <output>] [-h]
```

-a—All session statistics (active, stale, stopped)

-d—<Directory> Default: .

-f—<Filename> Default: 00\*.log

-v—verbose Default: off

-p—Clear phantom sessions

-o—<Filename> Output log to TEXT

-h—Help, usage

[Table 9-1](#) lists the options available with `carbs.pl` and their description.

**Table 9-1** Carbs.pl Options and Description

| Option        | Description  |
|---------------|--|
| -d<directory> | Optional. Accepts a directory as parameter with no trailing slash. You can use this option to change the default directory to scan for BackingStore log files. Default is current directory.   |
| -f<logfile>   | Optional. Accepts a logfile as parameter with no leading or trailing slashes. You can use this option to change the default log files. Allows you to enter individual logfile name as well as wildcard characters surrounded by single quotes. |
| -v            | Optional. No parameters. You can use this option to get total session count and phantom session count.   |
| -p            | Optional. No parameters. Generates a list of phantom sessions. You can use this option to clear the stale sessions.  |
| -o            | Optional. Accepts <output file> as parameter. You can use this option to convert BackingStore log files to readable files and write the results to the output file specified.  |
| -a            | Optional. No parameters. You can use this option to print all session statistics, such as per-NAS stale session count, total active sessions, and total stale sessions.  |
| -h            | You can use this option to get help with usage of carbs.pl.  |

## Configurable Worker Threads Enhancement

Prime Access Registrar provides a configurable variable you can use to increase the number of worker threads to handle a greater number of RADIUS packets during peak operating periods. This variable controls the processing of greater number of RADIUS packets than expected during peak operating periods.

The variable, RADIUS\_WORKER\_THREAD\_COUNT, is found in the **arserver** file under **/cisco-ar/bin/arserver** and controls the number of worker threads the Prime Access Registrar server creates. You can increase the number of worker threads to help make more efficient use of the server's CPU.



### Note

Before you increase the setting for RADIUS\_WORKER\_THREAD\_COUNT, you should be certain that you are running into a worker thread starvation issue. If you use scripts that consume a lot of processing and memory, you might run out of memory if you create too many worker threads.

Increasing the number of worker threads also increases memory utilization.

The purpose of this enhancement is to take advantage of spare CPU bandwidth which was not being used in earlier releases of Prime Access Registrar due to a lower number of worker threads. At times, the worker threads would be stuck doing work that took a long time to complete, like running a script. Having more threads will help mitigate these situations and will help improve on the latency created due to lack of free worker threads.

**Note**

Before modifying the RADIUS\_WORKER\_THREAD\_COUNT variable, consult with a TAC representative to ensure that modifying the RADIUS\_WORKER\_THREAD\_COUNT is warranted.

**Modifying the RADIUS WORKER THREAD COUNT**

To modify the RADIUS\_WORKER\_THREAD\_COUNT variable:

**Step 1** Log into the Prime Access Registrar server as a root user and change directory to **/cisco-ar/bin**.

**Step 2** Use a text editor and open the **arserver** file.

**Step 3** Locate the line with the RADIUS\_WORKER\_THREAD\_COUNT variable.

```
#change this to configure number of worker threads
RADIUS_WORKER_THREAD_COUNT=256
```

**Step 4** Modify the number of RADIUS worker threads to the number you choose.

**Note**

There is no upper limit to the number of RADIUS worker threads you can enable in your Prime Access Registrar server, but you should take care not to exceed your server's memory capacity.

**Step 5** Save the file and restart the Prime Access Registrar server.

## Session-Key Lookup

The Session-Key Lookup feature enables you to identify the Session Manager and Session Key of an existing session based on certain attributes associated with that session, such as the Mobile Station Integrated Services Digital Network (MSISDN) number.

The Session-Key Lookup feature requires the following enhancements to Prime Access Registrar software:

- Enabling a query service to be invoked for Ascend-IP-Allocate packets
- Enabling the setting of the Session-Key and Session-Manager environment variables by a query operation
- Performing session management after the query operation
- A new environment variable, Set-Session-Mgr-And-Key-Upon-Lookup, which when set to TRUE causes a session-cache Resource Manager to set the Session-Manager and Session-Key environment variables during the query lookup.

The Session-Key Lookup feature is useful in a scenario where an existing session requires an update from an incoming Ascend-IPA-Allocate packet (from a different NAS or device) with modified authorization attributes. Note that this Ascend-IPA-Packet might not have the exact set of attributes as

the original packet that created the session. However, the Ascend-IPA-Allocate packet must contain at least one attribute that can uniquely identify the session (such as the MSISDN number) and should contain the same UserName of the original session.

The Session-Key Lookup feature works in tandem with the RADIUS Query feature, where a RADIUS Query service is defined with the unique attribute (such as the MSISDN number) as the query-key and is configured to query all session managers. The Query-Service environment variable is set to the defined RADIUS Query service and the new environment variable (Set-Session-Mgr-And-Key-Upon-Lookup) is set to TRUE for this Ascend-IPA-Allocate packet. This triggers a query operation on all the live sessions. If there is a match, the Session-Manager and Session-Key of that session is used for subsequent session management. During session management, the session cache is updated with the modified authorization attributes.

The Session-Manager (or any outgoing script that executes after the Session-Manager Outgoing Script) should not reject the packet when doing a Session-Key lookup. Doing so causes the session to be deleted.

## Query-Notify

The Query-Notify feature enables you to store information about Wireless Application Protocol (WAP) gateways that have queried for User Identity-IP Address mapping and send appropriate messages to the WAP gateway when the subscriber logs out of the network.

Prime Access Registrar has been enhanced to update the session cache with the attribute-value pairs of an interim accounting update packet. This ensures the Prime Access Registrar server provides updated or current information to the WAP gateway during the proxy of interim records or query of the session cache.

Prime Access Registrar has been enhanced to also notify the WAP gateways that have queried a session with interim accounting update packets. If a WAP gateway does not respond to the Interim accounting update packets, the Prime Access Registrar server times out and retries by notifying the WAP gateways again. If there is no response after all the retries, the proxy packet is deleted and no change is made to the session or the WAP gateway's state in the Prime Access Registrar server. You can configure the number of retries under **/Radius/Clients/notificationproperties**.

The accounting response packet from the Prime Access Registrar server to the GPRS Gateway Support Node (GGSN) is independent of the proxy operation to the WAP gateways. The accounting response packet is sent back immediately without waiting for responses from the WAP gateways.

The Query-Notify feature also enables you to quarantine IP addresses for a configurable amount of time if a WAP gateway does not respond to Accounting-Stop packets sent by the Prime Access Registrar server.

The Prime Access Registrar server stores information about clients (usually the IP address) that queried for particular user information and sends RADIUS Accounting-Stop packets to those clients when the Prime Access Registrar server receives the Accounting-Stop packet. There is no intermediate proxy server between the Prime Access Registrar server and the WAP gateway.

To support the Query-Notify feature, the Prime Access Registrar server's *radius-query* service has been modified to also store information like the IP address about the clients queried for cached information. The information is stored in the user session record along with the cached information so it is available after a server reload.

### Configuring the Query-Notify feature

To configure the Query-Notify feature:

- 
- Step 1** Configure the Clients object under **/Radius/Clients**.
- Step 2** Set the EnableNotifications property to TRUE.
- The EnableNotifications property indicates that a client can receive Accounting-Stop notifications from the Prime Access Registrar server. When EnableNotifications is set to TRUE, a sub-directory named NotificationProperties appears in client object configuration.
- Step 3** Configure the properties under the client's NotificationProperties subdirectory.
- See , for information about how to configure these properties.
- Step 4** Configure a list of attributes to store under **/Radius/Advanced/Attribute Groups/<Notification Group>** where *<notification group>* is the name of an Attribute Group containing a list of attributes to be stored.
- 

This section contains the following topics:

- [Call Flow](#)
- [Configuration Examples](#)
- [Memory and Performance Impact](#)

## Call Flow

This section describes the call flow of the Query-Notify feature.

1. The Prime Access Registrar server caches information from an Accounting-Start. This information is usually from a GGSN when a subscriber enters into the network.
  2. When a WAP gateway receives a request to authenticate a subscriber, it queries the Prime Access Registrar server using an Access-Request packet to retrieve the cached information for that subscriber.
  3. The Prime Access Registrar server responds with Access-Accept if an entry is found for the subscriber in its cache; otherwise the server returns an Access-Reject. The Prime Access Registrar server sends an Access-Accept packet to the WAP gateway. The list of attributes sent in this Access-Accept will depend on radius-query service configuration.
- 
-  **Note** You use **aregcmd** to configure the attributes for the Access-Accept packet in the AttributesToBeReturned subdirectory under a radius-query service type.
- 
4. If the Prime Access Registrar server finds a cache entry for the subscriber and if the EnableNotifications property is set to TRUE, the Prime Access Registrar server stores the client IP address in the subscriber's cache.
  5. If the Prime Access Registrar server receives an Accounting-Interim-Update packet from the GGSN, it responds by sending an Accounting-Response packet then sends the Accounting-Interim-Update packets to all the queried clients of the WAP Gateways.

If the WAP gateway queried clients do not respond to the Accounting-Interim-Update packets, the Prime Access Registrar server times out and retries by notifying the WAP gateways again. If there is no response after all the retries, the proxy packet is deleted and no change is made to the session or the WAP gateway's state in the Prime Access Registrar server. The `StaleSessionTimeout` property under `/Radius/Advanced` is not applicable for Accounting-Interim-Update packets.

6. When the subscriber logs out of the network, the Prime Access Registrar server receives an Accounting-Stop packet and responds by sending an Accounting-Response back to the client.

Before releasing the subscriber's session, the Prime Access Registrar server looks for any client IP addresses in the subscriber's cache. If it finds any, the Prime Access Registrar server sends Accounting-Stop packets to those clients with the attributes configured in the `NotificationAttributeGroup` subdirectory for each client.

The Prime Access Registrar server forms the attributes with those attributes in the session cache and from the Accounting-Stop packet. The Prime Access Registrar server uses the value configured for the `Port` property in the `NotificationProperties` subdirectory as the destination port for the Accounting-Stop packet and uses the client's shared secret.

The Prime Access Registrar server then waits for Accounting-Response packets from each client to which it has sent Accounting-Stop packets. The Prime Access Registrar server waits for the time interval configured in the `InitialTimeout` property configured in the `NotificationProperties` subdirectory before sending another Accounting-Stop packet. If it does not receive an Accounting-Response packet, the Prime Access Registrar server sends additional Accounting-Stop packets until the number of attempts reaches the value configured in the `MaxTries` property in the `NotificationProperties` subdirectory.

7. When the Prime Access Registrar server receives an Accounting-Response packet from each client, the server releases the subscriber session.

If the Prime Access Registrar server does not receive Accounting-Response packets from all clients after the configured time and attempts, the server maintains the subscriber session for the time interval configured in the `StaleSessionTimeout` property in `/Radius/Advanced` then releases the subscriber session.

The Prime Access Registrar server maintains the subscriber session to address the quarantine IP address requirement. The Prime Access Registrar server must quarantine IP addresses if a WAP gateway does not respond to Accounting-Stop sent by the Prime Access Registrar server. The length of time an IP address is quarantined depends on the value of the `InitialTimeOut` property under the `NotificationProperties` subdirectory of `/Radius/Clients/wap_gateway`.

8. If the `StaleSessionTimeout` property is `TRUE` for a subscriber session, the Prime Access Registrar server rejects any query requests from clients for this session cache. After the `StaleSessionTimeout` expires, the Prime Access Registrar server will again send Accounting-Stop to all the clients listed in the session and proceeds to delete this subscriber session regardless of the status of the Accounting-Stop.

## Configuration Examples



### Note

In addition to the following configuration, the `StaleSessionTimeout` property must be set in `/Radius/Advanced`. This property has a default value of 1 hour.

The following shows an example configuration for a Query-Notify client:

```
[ //localhost/Radius/Clients/wap-gateway1 ]
Name = wap-gateway1
Description =
IPAddress = 10.100.10.1
SharedSecret = secret
Type = NAS
Vendor =
IncomingScript~ =
OutgoingScript~ =
EnableDynamicAuthorization = FALSE
NetMask =
EnableNotifications = TRUE
NotificationProperties/
  Port = 1813
  InitialTimeout = 5000
  MaxTries = 3
  NotificationAttributeGroup = notifyGroup
```

The following shows an example configuration for a Query-Notify AttributeGroup:

```
[ //localhost/Radius/Advanced/AttributeGroups/notifyGroup ]
Name = notifyGroup
Description =
Attributes/
  1. User-Name
  2. Acct-Session-Id
  3. NAS-Identifier
  4. NAS-Port
```

## Memory and Performance Impact

Using the Query-Notify feature will have the following effects:

- There will be a memory impact because the Prime Access Registrar server caches IP addresses of clients queried in the session record.
- There will be an impact on performance because the Prime Access Registrar server has to persist the cached IP address information before responding to **radius-query** requests.

## Support for Windows Provisioning Service

Prime Access Registrar supports Microsoft's Windows Provisioning Service (WPS). WPS provides hotspot users with seamless service to public WLAN hotspots by using Microsoft Windows-based clients. The Microsoft WPS solution requires Microsoft-based software in the data center for the RADIUS server and the provisioning server.

This section contains the following topics:

- [Call Flow](#)
- [Example Configuration](#)
- [Unsupported Features](#)

## Call Flow

The following is the WPS process and Wireless Internet Service Provider (WISP) packet sequence for a new wireless client login at a Wi-Fi hotspot location:

1. The client discovers the WISP network at a Wi-Fi hotspot.
2. The client authenticates as guest (with null username and credentials) to the Prime Access Registrar server .
3. The client is provisioned and a new account is created.
4. The client is authenticated using the new account credentials and accesses the Internet.

The Prime Access Registrar server performs the following functions during WPS:

1. Detects the guest subscriber login from the null username and null credentials during PEAPv0 (MS-PEAP) authentication.
2. Grants a successful login and returns a *sign-up* URL of the provisioning server as a PEAP-Type-Length-Value (TLV) in the next Access-Challenge Packet.

The following is an example value for the URL PEAP-TLV:

```
http://www.example.com/provisioning/master.xml#sign up
```

Where *#sign up* is the parameter for this action and is a required element of the value.

The sign-up URL value is passed when the user authenticates as guest. The sign-up URL is a fragment within the Master URL. You can also configure other fragments to be returned in the Master URL. See [Master URL Fragments, page 9-11](#) for more information about the different fragments.

3. Sends a VLAN-ID or IP filter (or both) in the final Access-Accept packet to restrict the guest user's accessibility to only the Provisioning server.
4. Authenticates using the user configuration in the user database after the client is provisioned and a new account is created.

## Example Configuration

The following shows an example configuration for the WPS feature:

```
[ //localhost/Radius/Services/peapv0 ]
  Name = peapv0
  Description =
  Type = peap-v0
  IncomingScript~ =
  OutgoingScript~ =
  MaximumMessageSize = 1024
  PrivateKeyPassword = <password>
  ServerCertificateFile = <path_to_ServerCertificateFile>
  ServerRSAKeyFile = <path_to_ServerRSAKeyFile>
  CACertificateFile = <path_to_CACertificateFile>
  CACertificatePath = <path_to_CACertificatePath>
  ClientVerificationMode = Optional
  VerificationDepth = 4
  EnableSessionCache = True
  SessionTimeout = "5 Minutes"
  AuthenticationTimeout = 120
  TunnelService = eap-mschapv2
  EnableWPS = True
  MasterURL = http://www.example.com/provisioning/master.xml
```

```
WPSGuestUserProfile = WPS-Guest-User-Profile
```

When you set the EnableWPS property to TRUE, you must provide values for the properties MasterURL and WPSGuestUserProfile. See [Environment Variables, page 9-11](#) for more information.

## Environment Variables

The following two environment variables are used to support WPS:

- [Send-PEAP-URI-TLV](#)
- [Master-URL-Fragment](#)

### Send-PEAP-URI-TLV

Send-PEAP-URI-TLV property is a Boolean value used by the authenticating user service to make the PEAP-V0 service include the URI PEAP-TLV in the protected success message. Under different circumstances Prime Access Registrar might send back different fragments within the MasterURL to the client, as described above.

The conditions under which this has to be sent is best known to the user authentication service (the service that is specified within the eap-mschapv2 service, which in turn is the tunnel service for PEAP-V0 service). So when it decides that it needs to send back the URL it can set this variable to TRUE. The default value for this is FALSE.

### Master-URL-Fragment

The Prime Access Registrar authenticating user service uses Master-URL-Fragment to set the fragment within the Master URL that needs to be sent back. The Prime Access Registrar user authentication service sets the fragment to different values under different circumstances. While the Send-PEAP-URL-TLV indicates whether to send the URL or not, Master-URL-Fragment is used to intimate which fragment within the URL needs to be sent. If this variable is not set and if it is required to send the URL, '#signup' will be sent by default.

## Master URL Fragments

The following sections describe the different fragments the RADIUS server might send to the AP in the Master URL:

- [Sign up](#)
- [Renewal](#)
- [Password change](#)
- [Force update](#)

### Sign up

This value is passed when the user authenticates as guest. The following is an example value for the URL PEAP-TLV:

```
http://www.example.com/provisioning/master.xml#sign up
```

where #sign up is the parameter for this action and a required element of the value.

## Renewal

This value is passed when the user's account is expired and needs renewal before network access can be granted. The following is an example value for the URL PEAP-TLV:

```
http://www.example.com/provisioning/master.xml#renewal
```

where #renewal is the parameter for this action and a required element of the value.

## Password change

This value is passed when the user is required to change the account password. An example value for the URL PEAP-TLV is:

```
http://www.example.com/provisioning/master.xml#passwordchange
```

where #passwordchange is the parameter for this action and a required element of the value.

## Force update

This value is passed when the WISP requires the Wireless Provisioning Services on the client to download an updated XML master file. This method of updating the XML master file on the client should be used only to correct errors; otherwise, the TTL expiry time in the XML master file is used to provide background updates. The following is an example value for the URL PEAP-TLV:

```
http://www.example.com/provisioning/master.xml#forceupdate
```

where #forceupdate is the parameter for this action and a required element of the value.

## Unsupported Features

The following features are part of the Microsoft WPS functionality, but are not supported in the Prime Access Registrar:

- [Account Expiration and Renewal](#)
- [Password Changing and Force Update](#)

## Account Expiration and Renewal

When the user creates an account and logs in with that account, the RADIUS server authenticates and authorizes the request and sends back an Access-Accept with a Session-Timeout attribute. The Access Point (AP) then forces the wireless client to reauthenticate for every timeout value. When there is one timeout duration left in the user account, the RADIUS server needs to send back a *renewal* URL (a URL fragment within the master URL) to the client for the user to renew the account.

Prime Access Registrar does not support this feature because the interface the Prime Access Registrar server has with the CiscoSecure Remote Agent does not have provisions to get the expiration information of user account. However, this release does provide an environment variable to copy the URL fragment and to control whether or not to send the URL using another environment variable. This can be used to send the renewal URL. There are some limitations, however.

## Password Changing and Force Update

The Password Changing option is passed when the user is required to change the account password. Force Update option is passed when the WISP requires the Wireless Provisioning Services on the client to download an updated XML master file.

These functions are not possible in this release for the same reason mentioned above, the loose coupling between Prime Access Registrar and the CiscoSecure Remote Agent. Additionally, there is no known use case for this. As mentioned above, you can use the newly added environment variables to trigger these options.

## Command Completion

Prime Access Registrar's command completion feature provides online help by listing possible entries to the current command line when you press the Tab key after entering a partial command. The Prime Access Registrar server responds based on:

- The location of the cursor including the current directory
- Any data you have entered on the command line prior to pressing the Tab key

The command completion feature emulates the behavior of Cisco IOS and Kermit. When you press the Tab key after entering part of a command, the Prime Access Registrar server provides any identifiable object and property names. For example, after you first issue **aregcmd** and log into Prime Access Registrar, enter the following:

```
cd <Tab>
```

```
Administrators/ Radius/
```

Pressing the Tab key consecutively displays possible context-sensitive choices.

In the following example, after changing directory to **/Radius/services/local-file** an administrator wants to see the possible types of authentication services that can set.

```
cd /Radius/services/local-file
```

```
//localhost/Radius/Services/local-file ]
Name = local-file
Description =
Type = file
IncomingScript~ =
OutgoingScript~ =
OutagePolicy~ = RejectAll
OutageScript~ =
FilenamePrefix = accounting
MaxFileSize = "10 Megabytes"
MaxFileAge = "1 Day"
RolloverSchedule =
```

```
set type <Tab>
```

```
eap-leap      file      local      radius-session
eap-md5       group    odbc       rex
eap-sim       ldap     radius     tacacs-udp
```

Values can also be tab-completed. For example, if you decide to set the local-file service's type to file, you can do the following:

```
set type f<Tab>
```

and the command line completes to:

```
set type file
```

## Service Grouping Feature

The Service Grouping feature enables you to specify multiple services (called *subservices*) to be used with authentication, authorization, or accounting requests. The general purpose is to enable multiple Remote Servers to process requests.

Perhaps the most common use of this feature will be to send accounting requests to multiple Remote Servers thus creating multiple accounting logs. Another common use might be to authenticate from more than one Remote Server where, perhaps the first attempt is rejected, other Remote Servers can be attempted and an Access-Accept obtained.

Clearly, in the accounting request example, each request must be successfully processed by each subservice in order for the originator of the accounting request to receive a response. This is known as a *logical AND* of each of the subservice results. In the authenticate example, the first subservice which responds with an accept is returned to the client or if all subservices respond with *reject*, then a reject is returned to the client. This is known as a *logical OR* of each of the subservice results.

A Service is specified as a Group Service by setting its type to *group*, specifying the ResultRule (AND or OR) and specifying one or more subservices in the GroupServices subdirectory. The subservices are called in numbered order and as such are in an indexed list similar to Remote Server specification in a radius Service. Incoming and outgoing scripts for the Group Service can be optionally specified.

A subservice is any configured non-Group Service. When a Group Service is used, each subservice is called in exactly the same manner as when used alone (such as if specified as the DefaultAuthenticationService). Incoming and Outgoing scripts are executed if configured and Outage Policies are honored.

This section contains the following topics:

- [Configuration Example - AccountingGroupService](#)
- [Configuration Example 2 - AuthenticationGroupService](#)

### Configuration Example - AccountingGroupService

To configure an accounting Group Service to deliver accounting requests to multiple Remote Servers:

---

**Step 1** The first task is to set up the subservices which are to be part of the AccountingGroupService. Since subservices are merely configured Services which have been included in a service group, you need only define two new Services.

For this example, we will define two new radius Services called *OurAccountingService* and *TheirAccountingService*. A provider might want to maintain duplicate accounting logs in parallel with their bulk customer's accounting logs.

**Step 2** Change directory to **/radius/services**. At the command line, enter the following:

**cd /radius/services**

```
[ //localhost/Radius/Services ]
Entries 1 to 2 from 2 total entries
Current filter: <all>
local-file/
local-users/
```

**Step 3** At the command line, enter the following:

```
add OurAccountingService
```

```
add TheirAccountingService
```

The configuration of these Services is very similar to standalone Radius accounting service. Step-by-step configuration instructions are not provided, but the complete configuration is shown below:

```
[ //localhost/Radius/Services/OurAccountingService ]
Name = OurAccountingService
Description =
Type = radius
IncomingScript = OurAccountingInScript
OutgoingScript = OurAccountingOutScript
OutagePolicy = RejectAll
OutageScript =
MultipleServersPolicy = Failover
RemoteServers/
  1. OurPrimaryServer
  2. OurSecondaryServer

[ //localhost/Radius/Services/TheirAccountingService ]
Name = TheirAccountingService
Description =
Type = radius
IncomingScript = TheirAccountingInScript
OutgoingScript = TheirAccountingOutScript
OutagePolicy = RejectAll
OutageScript =
MultipleServersPolicy = Failover
RemoteServers/
  1. TheirPrimaryServer
  2. TheirSecondaryServer
```

The next step is to create the new **AccountingGroupService**. The purpose of this Service is to process Accounting requests through both **OurAccountingService** and **TheirAccountingService**.

**Step 4** At the command line, enter the following:

```
add AccountingGroupService
```

```
Added AccountingGroupService
```

```
cd AccountingGroupService
```

```
[ //localhost/Radius/Services/AccountingGroupService ]
Name = AccountingGroupService
Description =
Type =
IncomingScript =
OutgoingScript =
```

**set type group**

```
Set Type group
```

**Step 5** Set the ResultRule to *AND* to ensure that both services process the accounting request successfully.

**set ResultRule AND**

```
Set ResultRule AND
```

**Is**

```
[ //localhost/Radius/Services/AccountingGroupService ]
Name = AccountingGroupService
Description =
Type = group
IncomingScript =
OutgoingScript =
ResultRule = AND
GroupServices/
```

**set IncomingScript AcctGroupSvcInScript****set OutgoingScript AcctGroupSvcOutScript**

Add OurAccountingService and TheirAccountingService as subservices of the Group Service.

**Step 6** At the command line, enter the following:

**cd GroupServices**

```
[ //localhost/Radius/Services/AccountingGroupService/GroupServices ]
```

**set 1 OurAccountingService**

```
Set 1 OurAccountingService
```

**Set 2 TheirAccountingService**

```
Set 2 TheirAccountingService
```

**Is**

```
[ //localhost/Radius/Services/AccountingGroupService ]
Name = AccountingGroupService
Description =
Type = group
IncomingScript = AcctGroupSvcInScript
OutgoingScript = AcctGroupSvcOutScript
ResultRule = AND
GroupServices/
  1. OurAccountingService
  2. TheirAccountingService
```

This completes the setup of the AccountingGroupService. To use this Service simply set it as the DefaultAccountingService and/or configure a policy/rule set which will select this Service. Essentially, this can be used in the same manner as any other standalone service.

## Summary of Events

The following describes the flow of what happens when a client sends an accounting request which is processed by the AccountingGroupService:

1. ActGroupSvcInScript is executed.
2. OurAccountingService is called.
3. OurAccountingService's Incoming Script, OurAccountingInScript is called.
4. The request is sent to the Remote Server OurPrimaryServer and/or OurSecondaryServer, if necessary.
5. If a response is not received, because we used the **AND** ResultRule, the request failed and no response is sent to the client and the request is dropped. If a response is received, then the process continues.
6. OurAccountingService's Outgoing Script, OurAccountingOutScript is called.
7. TheirAccountingService is called.
8. TheirAccountingService's Incoming Script, TheirAccountingInScript is called.
9. The request is sent to the Remote Server TheirPrimaryServer and/or TheirSecondaryServer, if necessary.
10. If a response is not received, because we used the **AND** ResultRule, the request failed and no response is sent to the client and the request is dropped. If a response is received, then the process continues.
11. TheirAccountingService's Outgoing Script, TheirAccountingOutScript is called.
12. AcctGroupSvcOutScript is executed.
13. Standard processing continues.

## Configuration Example 2 - AuthenticationGroupService

To configure a Group Service for the purposes of providing alternate Remote Servers for a single authentication:



### Note

If Service A rejects the request, try Service B.

### Step 1

The first task is to set up the subservices which are to be part of the AuthenticationGroupService. Since subservices are merely configured Services which have been included in a service group, we will simply define two new Services. For simplicity, we will define two new radius Services called AuthenticationServiceA and AuthenticationServiceB.

### Step 2

At the command line, enter the following:

```
cd /radius/services

[ //localhost/Radius/Services ]
```

```

Entries 1 to 2 from 2 total entries
Current filter: <all>
local-file/
local-users/

```

**add AuthenticationServiceA**

**add AuthenticationServiceB**

**Step 3** The configuration of these Services is very similar to standalone Radius authentication service. Step-by-step configuration instructions are not provided, but the complete configuration is shown below:

```

[ //localhost/Radius/Services/AuthenticationServiceA ]
  Name = AuthentictionServiceA
  Description =
  Type = radius
  IncomingScript = AuthAInScript
  OutgoingScript = AuthAOutScript
  OutagePolicy = RejectAll
  OutageScript = AuthAOutageScript
  MultipleServersPolicy = Failover
  RemoteServers/
    1. PrimaryServerA
    2. SecondaryServerA

[ //localhost/Radius/Services/AuthenticationServiceB ]
  Name = AuthentictionServiceB
  Description =
  Type = radius
  IncomingScript = AuthBInScript
  OutgoingScript = AuthBOutScript
  OutagePolicy = RejectAll
  OutageScript = AuthBOutageScript
  MultipleServersPolicy = Failover
  RemoteServers/
    1. PrimaryServerB
    2. SecondaryServerB

```

The next step is to create the new "AuthenticationGroupService". The purpose of this Service is to process authentication requests through both AuthenticationServiceA and AuthenticationServiceB if AuthenticationServiceA rejects the request.

**Step 4** At the command line, enter the following:

**add AuthenticationGroupService**

```
Added AuthenticationGroupService
```

**cd AuthenticationGroupService**

```
[ //localhost/Radius/Services/AuthenticationGroupService ]
  Name = AuthenticationGroupService
  Description =
  Type =
  IncomingScript =
  OutgoingScript =
```

**set type group**

```
Set Type group
```

Next set the ResultRule to **OR** because we want to ensure that if the first subservice rejects the request, we then try the second subservice. If the second subservice rejects the request, then the response to the client is a reject.

**Step 5** At the command line, enter the following:

**set ResultRule OR**

```
Set ResultRule OR
```

**Set IncomingScript AuthGroupSvcInScript**

```
Set OutgoingScript AuthGroupSvcOutScript
```

**Set IncomingScript AuthGroupSvcInScript**

```
Set OutgoingScript AuthGroupSvcOutScript
```

**ls**

```
[ //localhost/Radius/Services/AuthenticationGroupService ]
  Name = AuthenticationGroupService
  Description =
  Type = group
  IncomingScript = AuthGroupSvcInScript
  OutgoingScript = AuthGroupSvcOutScript
  ResultRule = OR
  GroupServices/
```

Now we must add the services we created "AuthenticationServiceA" and "AuthenticationServiceB" as subservices of the Group Service.

**Step 6** At the command line, enter the following:

**cd GroupServices**

```
[ //localhost/Radius/Services/AuthenticationGroupService/GroupServices ]
```

**set 1 AuthenticationServiceA**

```
Set 1 AuthenticationServiceA
```

**Set 2 AuthenticationServiceB**

```
Set 2 AuthenticationServiceB
```

**Is**

```
[ //localhost/Radius/Services/AuthenticationGroupService ]
  Name = AuthenticationGroupService
  Description =
  Type = group
  IncomingScript = AuthGroupSvcInScript
  OutgoingScript = AuthGroupSvcOutScript
  ResultRule = OR
  GroupServices/
    1. AuthenticationServiceA
    2. AuthenticationServiceB
```

---

This completes the setup of the AuthenticationGroupService. To use this Service simply set it as the DefaultAuthenticationService and/or configure a policy/rule set which will select this Service. Essentially, this can be used in the same manner as any other standalone Service.

**Summary of Events**

The following describes the flow of what happens when a client sends an Authentication request which is processed by the AuthenticationGroupService:

1. AuthGroupSvcInScript is executed.
2. AuthenticationServiceA is called.
3. AuthenticationServiceA's Incoming Script, AuthAInScript is called.
4. If the response is a reject or the request is dropped (due to an Outage Policy):
  - a. AuthenticationServiceA's Outgoing Script, AuthAOutScript is called.
  - b. Processing continues with the next service.
5. If the response is an Accept:
  - a. AuthenticationServiceA's Outgoing Script, AuthAOutScript is called.
  - b. Skip to step 9.
6. AuthenticationServiceB is called.
7. AuthenticationServiceB's Incoming Script, AuthBInScript is called.
8. Since this is the last subservice in our Group Service:
  - a. AuthenticationServiceB's Outgoing Script, AuthBOutScript is called.

- b. Regardless of whether the request is Accepted or Rejected, processing will continue at step 9.
9. AuthGroupSvcOutScript is executed.
10. Standard processing continues.

## SHA-1 Support for LDAP-Based Authentication

The Prime Access Registrar server supports secure hash algorithm (SHA-1) for LDAP-based authentication. This feature enables the Prime Access Registrar server to authenticate users whose passwords are stored in LDAP servers and hashed using the SHA-1 encoding scheme.

SHA-1 support actually adds functionality for the following three features to Prime Access Registrar:

- Authentication of PAP access requests against an LDAP user entry that uses the SHA-algorithm to the hash password attribute
- Authentication of PAP access requests against an LDAP user entry that uses the SSHA algorithm to hash the password attribute
- Configuration of the Prime Access Registrar server to dynamically determine how password attributes retrieved from LDAP are encrypted and process them accordingly

This enhancement is 100% backwards compatible. All previously supported values for the PasswordEncryptionStyle property are still supported and still provide the same behavior. The only noticeable change is that **dynamic** is now the default value for the PasswordEncryptionStyle property.

This section contains the following topics:

- [Remote LDAP Server Password Encryption](#)
- [Dynamic Password Encryption](#)
- [Logs](#)

## Remote LDAP Server Password Encryption

Apart from the two values, none and crypt, of the **PasswordEncryptionStyle** property on a Remote LDAP Server, SHA-1 supports adds three additional values for the PasswordEncryptionStyle property. [Table 9-2](#) lists the valid values for this property and describes the corresponding behavior.

**Table 9-2 Remote LDAP Server Password Encryption Style Values**

| PasswordEncryptionStyle | Cisco Prime Access Registrar Behavior  |
|-------------------------|--|
| none                    | All passwords retrieved from this LDAP server are assumed to be returned to Prime Access Registrar as clear text. (There is no change in this functionality.)  |
| crypt                   | All passwords retrieved from this LDAP server are assumed to be returned to Prime Access Registrar as passwords encrypted using the UNIX <i>crypt</i> algorithm. (There is no change in this functionality.)<br><br>Passwords can be preceded by the {crypt} prefix, which is stripped before comparing passwords. |

**Table 9-2 Remote LDAP Server Password Encryption Style Values (continued)**

| <b>PasswordEncryptionStyle</b> | <b>Cisco Prime Access Registrar Behavior</b>  |
|--------------------------------|---|
| SHA-1                          | <p>All passwords retrieved from this LDAP server are assumed to be returned to Prime Access Registrar as a Base64-encoded version of the user's password after it has been hashed using the SHA-1 mechanism (as defined by Netscape).</p> <p>Passwords can be preceded by the {sha} prefix, which is stripped before comparing passwords.</p>   |
| SSHA-1                         | <p>All passwords retrieved from this LDAP server are assumed to be encrypted/hashed using the SSHA mechanism (as defined by Netscape). Passwords can be preceded by the {ssha} prefix, which is stripped before comparing passwords.</p> <p><b>Note</b> This is a Netscape/iPlanet-specific mechanism.</p>  |
| EAP-Mschapv2                   | <p>All passwords received from the LDAP server are expected to be returned to Prime Access Registrar as NT LAN Manager (NTLM) V1 hashes using the MD4 algorithm (RFC1320). NTLM v1 hashes are generated from the clear text password provided by the user. The NTLM passwords are stored with an <b>NTLMv1=</b> prefix in the database as shown in the example below.</p> <p>Example: NTLMv1=5B3844FB41E27C48A93B6C8C6864FB83</p> <p>This password encryption style is also applicable for Oracle-based authentication.</p> |
| dynamic                        | <p>The value instructs Prime Access Registrar to choose the encryption mechanism on a case-by-case basis after it determines the presence of a known prefix, which the LDAP server prepends to the value of the password attribute.</p> <p>For example, if the following was returned from an LDAP server as a password attribute: {SHA}qZk+NkcGgWq6PiVxeFDCbJzQ2J0=, the password would be processed using the SHA-1 mechanism. This value will be the new default for the PasswordEncryptionStyle property.</p>           |

## Dynamic Password Encryption

When using the dynamic setting for the PasswordEncryptionStyle property on a Remote LDAP Server, the Prime Access Registrar server looks for the prefixes listed in [Table 9-3](#) to determine if encryption or a hash algorithm should be used during password comparison.



### Note

Password prefixes are not case-sensitive.

**Table 9-3 Remote LDAP Server Password Prefix Values**

| Password Prefix | Encryption/Hash Algorithm Used  |
|-----------------|---|
| none            | None; when no known prefix is found, the password attribute is assumed to be in clear text. |
| {crypt}         | UNIX crypt algorithm  |
| {sha}           | Secure Hash Algorithm, version 1 (SHA-1)  |
| {ssha}          | SSHA-1, as defined by Netscape.   |
| {NTLMv1}        | MD4 algorithm (RFC1320).  |

The default value for the PasswordEncryptionStyle property on a Remote LDAP Server is **dynamic**.

**Note**

Using the *dynamic* setting for the PasswordEncryptionStyle property will require a bit more processing for each password comparison. When using dynamic, the Prime Access Registrar server must examine each password for a known prefix. This should have no visible impact on performance.

## Logs

Turn on trace to level 4 to indicate (via the trace log) which password comparison method is being used.

## Dynamic Attributes

Prime Access Registrar supports dynamic values for the configuration object properties listed below. Dynamic attributes are similar to UNIX shell variables. With dynamic attributes, the value is evaluated at run time. All of the objects that support dynamic attributes will have validation turned off in **aregcmd**.

This section contains the following topics:

- [Object Properties with Dynamic Support](#)
- [Dynamic Attribute Format](#)
- [Configuration](#)
- [Example](#)
- [Notes](#)
- [Validation](#)

## Object Properties with Dynamic Support

The following object properties support dynamic values:

Radius

DefaultAuthenticationService

DefaultAuthorizationService

DefaultAccountingService

DefaultSessionManager  
IncomingScript  
OutgoingScript



---

**Note** Do not use the following environment variables:  
Accounting-Service for the **/Radius/DefaultAccountingService**, Authentication-Service for the **/Radius/DefaultAuthenticationService**, or Authorization-Service for the **/Radius/DefaultAuthorizationService**  
User-Profile for the **BaseProfile**, User-Group for the **Group**, User-Authorization for the **AuthorizationScript**, Session-Manager for the **DefaultSessionManager**, or Session-Service for the **DefaultSessionService**.

---

/Radius/Clients  
    client1/  
        IncomingScript  
        OutgoingScript  
/Radius/Userlist/Default  
    user1/  
        Group  
        BaseProfile  
        AuthenticationScript  
        AuthorizationScript  
/Radius/UserGroup  
    Group1/  
        BaseProfile  
        AuthenticationScript  
        AuthorizationScript  
/Radius/Vendor  
    Vendor1/  
        IncomingScript  
        OutgoingScript  
/Radius/Service  
    Service1/  
        IncomingScript  
        OutgoingScript  
        OutageScript  
        OutagePolicy  
/Radius/RemoteServers  
    remoteserver1/  
        IncomingScript

```

OutgoingScript
Remoteldapservers1/
Searchpath
Filter

```



**Note** To differentiate the properties that support dynamic attributes, we place a tilde (~) after each property, as in IncomingScript~. However, when the Prime Access Registrar administrator is required to set values for those properties, continue to use the original property name, such as set IncomingScript \${elrealm}{Test}. The tilde is only for visual effect, and including the tilde will generate an error (“310 command Failed.”)

## Dynamic Attribute Format

The format of the dynamic attribute is:

```

${eq|attribute-name}{default-name}

```

where **e** stands for environment dictionary, **q** stands for request dictionary, and **p** stands for response dictionary. You can use e, q, and p in any order. The attribute name is the name for the attribute from environment dictionary, request dictionary, or response dictionary.

For example,

```

/Radius
DefaultAuthenticationService = ${eq|realm}{local-users}

```

The default Authentication Service is determined at run time. Prime Access Registrar first checks to see if there is one value of **realm** in the environment dictionary. If there is, it becomes the value of DefaultAuthenticationService. If there is not, check the value of realm in the request dictionary. If there is one value, it becomes the value of DefaultAuthenticationService. Otherwise, local-users is the DefaultAuthenticationService. If we do not set local-users as the default value, the DefaultAuthenticationService is *null*. The same concept applies to all other attribute properties.

The validation for the dynamic values of the object property will only validate the default value. In the above example, Prime Access Registrar will do validation to check whether local-users is one of services defined in the service subdirectory.



**Note**

When setting specific property values, do not use the tilde (~) in the property name. Doing so generates a *310 Command Failed* error.

## Tunneling Support Feature

Tunneling support is strictly based upon the IETF RFC: “RADIUS Attributes for Tunnel Protocol Support” (<http://www.ietf.org/rfc/rfc2868.txt>).

Table 9-4 lists the tunneling attributes supported in this Prime Access Registrar release.

**Table 9-4 Tunneling Attributes Supported by Prime Access Registrar**

| Attribute Number | Attribute               |
|------------------|-------------------------|
| 64               | Tunnel-Type             |
| 65               | Tunnel-Medium-Type      |
| 66               | Tunnel-Client-Endpoint  |
| 67               | Tunnel-Server-Endpoint  |
| 69               | Tunnel-Password         |
| 81               | Tunnel-Private-Group-ID |
| 82               | Tunnel-Assignment-ID    |
| 83               | Tunnel-Preference       |
| 90               | Tunnel-Client-Auth-ID   |
| 91               | Tunnel-Server-Auth-ID   |

The tunneling attribute has the following format:

| (1 byte) | (1 byte) | (1 byte) | (variable number of bytes) |
|----------|----------|----------|----------------------------|
| Type     | Length   | Tag      | Value                      |

This section contains the following topics:

- [Configuration](#)
- [Example](#)
- [Notes](#)
- [Validation](#)

## Configuration

1. Configure the tag attributes as untagged attributes under the **/Radius/Advanced/Attribute Dictionary** directory (for example, **Tunnel-Type**).
2. Attach the “**\_tag**” tag to these attributes when configuring the attributes under all of the other directories as tagged attributes (for example, **Tunnel-Type\_tag10** under the **/Radius/Profiles/test** directory). Without the tag number, the default value is (**\_tag = \_tag0**).

## Example

```

/Radius/Advanced/Attribute Dictionary
  /Tunnel-Client-ID
    Name = Tunnel-Client-Endpoint
    Description =
    Attribute = 66
    Type = STRING
    Min = 0
    Max = 253

/Radius/Profiles/test

```

```
Name = test
Description =
/Attributes
  Tunnel-Client-Endpoint_tag3 = "129.56.112.1"
```

## Notes

1. “\_tag” is reserved for the tunneling attributes. No other attributes should include this suffix.
2. The tag number value can range from 0 through 31.

## Validation

The Prime Access Registrar server checks whether the tag attributes are defined under the **/Radius/Advanced/Attribute Dictionary** directory. The server also checks whether the tag number falls within the range (0-31).

## xDSL VPI/VCI Support for Cisco 6400

To provide this support, a distinction must be made between device authentication packets and regular user authentication packets. This section contains the following topics:

- [Using User-Name/User-Password for Each Cisco 6400 Device](#)
- [Format of the New User-Name Attribute](#)

## Using User-Name/User-Password for Each Cisco 6400 Device

This approach assumes that for every 6400 NAS, a device-name/device-password is created for each. Following are the required changes:

For each NAS in Prime Access Registrar:

```
Name = test6400-1
Description =
IPAddress = 209.165.200.224
SharedSecret = secret
Type = NAS
Vendor =
IncomingScript =
OutgoingScript =
Device-Name = theDevice
Device-Password = thePassword
```

When the 6400 sends out the device authentication packet, it might have different **User-Name/User-Password** attributes for each 6400 NAS. When Prime Access Registrar receives the packet, it tries to obtain the **Device-Name/Device-Password** attributes from the NAS entry in the Prime Access Registrar configuration database. When the **User-Name/User-Password** in the packet match the configured **Device-Name/Device-Password** attribute values, Prime Access Registrar assumes that it must get the device. The next step is to replace the **User-Name** attribute with the concatenated `<module>/<slot>/<port>` string. From this point, the packet is treated as a regular packet.

**Note**

A user record with the name of the concatenated string must be created.

## Format of the New User-Name Attribute

After the device is identified, the **User-Name** attribute is replaced with the new value. This new value is the concatenation of 6400 <module>/<slot>/<port> information from the **NAS-Port** attribute and the packet is treated as a regular user authentication from this point on.

**Note**

This format only supports NAS Port Format D. See Cisco IOS documentation for more information about NAS port formats.

The format of the new **User-Name** attribute is the **printf** of “%s-%d-%d-%d-%d-%d” for the following values:

**NAS-IP**—in dot format of the **NAS-IP-Address** attribute. For example, 10.10.10.10.

**slot**—apply mask 0xF0000000 on **NAS-Port** attribute and shift right 28 bits. For example, **NAS-Port** is 0x10000000, the slot value is 1.

**module**—apply mask 0x08000000 on **NAS-Port** attribute and shift right 27 bits. For example, **NAS-Port** is 0x08000000, the module value is 1.

**port**—apply mask 0x07000000 on **NAS-Port** attribute and shift right 24 bits. For example, **NAS-Port** is 0x06000000, the port value is 6.

**VPI**—apply mask 0x00FF0000 on **NAS-Port** attribute and shift right 16 bits. For example, **NAS-Port** is 0x00110000, the VPI value is 3.

**VCI**—apply mask 0x0000FFFF on **NAS-Port** attribute. For example, **NAS-Port** is 0x00001001, the VCI value is 9.

## Apply Profile in Cisco Prime Access Registrar Database to Directory Users

You can define the **User-Profile** and **User-Group** environment variables in the directory mapping and Prime Access Registrar will apply the profiles defined in the Prime Access Registrar database to each directory user having any of these two variables set.

This section contains the following topics:

- [User-Profile](#)
- [User-Group](#)
- [Example User-Profile and User-Group Attributes in Directory User Record](#)

### User-Profile

This attribute is of type string with the format:

<Value1>::<Value2> ...

The **User-Profile** attribute is intended to hold a list of profile names. *<Value1>* and *<Value2>* represent the names of the profiles. They are separated by the “:” character, therefore, the “:” can not be part of the profile name. The order of values in the string has significance, as the profiles are evaluated from left to right. In this example, profile *<Value2>* is applied after profile *<Value1>*.

Assume the user record has a field called `UserProfile` that holds the name of the profile that applies to this user. This field is mapped to the environment attribute **User-Profile**. Following is how the mapping is done with **aregcmd**:

```
QuickExample/
  Name = QuickExample
  Description =
  Protocol = ldap
  IPAddress = 209.165.200.224
  Port = 389
  ReactivateTimerInterval = 300000
  Timeout = 15
  HostName = QuickExample.company.com
  BindName =
  BindPassword =
  UseSSL = FALSE
  SearchPath = "o=Ace Industry, c=US"
  Filter = (uid=%s)
  UserPasswordAttribute = password
  LimitOutstandingRequests = FALSE
  MaxOutstandingRequests = 0
  MaxReferrals = 0
  ReferralAttribute =
  ReferralFilter =
  PasswordEncryptionStyle = None
  LDAPToEnvironmentMappings/
    UserProfile = User-Profile
  LDAPToRadiusMappings/
```

After Prime Access Registrar authenticates the user, it checks whether **User-Profile** exists in the environment dictionary. If it finds **User-Profile**, for each value in **User-Profile**, Prime Access Registrar looks up the profile object defined in the configuration database and adds all of the attributes in the profile object to the response dictionary. If any attribute is included in more than one profile, the newly applied profile overrides the attribute in the previous profile.

## User-Group

You can use the **User-Group** environment variable to apply the user profile as well. In Prime Access Registrar, a user can belong to a user group, and that user group can have a pointer to a user profile. When Prime Access Registrar finds that a packet has **User-Group** set, it obtains the value of the **User-Profile** within the user group, and if the **User-Profile** exists, it applies the attributes defined in the user profile to that user.

Note that in Prime Access Registrar, every user can also directly have a pointer to a user profile. Prime Access Registrar applies profiles in the following order:

1. If the user profile defined in the user group exists, apply it.
2. If the user profile defined in the user record exists, apply it.

The profile in **User-Group** is more generic than in **User-Profile**. Therefore, Prime Access Registrar applies the profile from generic to more specific.

## Example User-Profile and User-Group Attributes in Directory User Record

You can use an existing user attribute in the user record to store profile info. When this is a new attribute, we suggest you create a new auxiliary class **AR\_UserRecord** for whichever user class is used.

**AR\_User\_Profile** and **AR\_User\_Group** are two optional members in this class. They are of type string. The mapping is as follows:

```
LDAPToEnvironmentMappings/
  AR_User_Profile = User-Profile
  AR_User_Group = User-Group
```

## Directory Multi-Value Attributes Support

If any attributes mapped from the LDAP directory to the Prime Access Registrar response dictionary are multivalued, the attributes are mapped to multiple RADIUS attributes in the packet.

## MultiLink-PPP (ML-PPP)

Prime Access Registrar supports MultiLink-PPP (ML-PPP). ML-PPP is an IETF standard, specified by RFC 1717. It describes a Layer 2 software implementation that opens multiple, simultaneous channels between systems, providing additional bandwidth-on-demand, for additional cost. The ML-PPP standard describes how to split, recombine, and sequence datagrams across multiple B channels to create a single logical connection. The multiple channels are the ports being used by the Network Access Server (NAS).

During the AA process, Prime Access Registrar authenticates the user connection for each of its channels, even though they belong to the same logical connection. The Authentication process treats the multilink connection as if it is multiple, single link connections. For each connection, Prime Access Registrar creates a session dedicated for management purposes. The session stays active until you logout, which subsequently frees up all of the ports in the NAS assigned to each individual session, or until the traffic is lower than a certain threshold so that the secondary B channels are destroyed thereafter. Prime Access Registrar has the responsibility of maintaining the active session list and discards any session that is no longer valid in the system, by using the accounting stop packet issued from NAS. The multiple sessions that were established for a single logical connection must be destroyed upon the user logging out.

In addition, the accounting information that was gathered for the sessions must be aggregated for the corresponding logical connection by the accounting software. Prime Access Registrar is only responsible for logging the accounting start and accounting stop times for each session. As those sessions belong to the same bundle, IETF provides two standard RADIUS attributes to identify the related multilink sessions. The attributes are **Acct-Multi-Session-Id** (attribute **50**) and **Acct-Link-Count** (attribute **51**), where **Acct-Multi-Session-Id** is a unique Accounting identifier used to link multiple related sessions in a log file, and **Acct-Link-Count** provides the number of links known to have existed in a given multilink session at the time the Accounting record was generated. The Accounting software is responsible for calculating the amount of the secondary B channel's connection time.

The secondary B channel can go up and down frequently, based upon traffic. The Ascend NAS supports the **Target-Util** attribute, which sets up the threshold for the secondary channel. When the traffic is above that threshold the secondary channel is up, and when the traffic is below that threshold, the secondary B channel is brought down by issuing an Accounting stop packet to Prime Access Registrar. On the other hand, if you bring down the primary channel (that is, log out), the secondary B channel is also destroyed by issuing another Accounting stop packet to Prime Access Registrar.

Table 9-5 lists ML-PPP related attributes.

**Table 9-5 ML-PPP Attributes**

| Number | Attribute             | Cisco NAS (IOS 11.3 Release) | Ascend NAS |
|--------|-----------------------|------------------------------|------------|
| 44     | Acct-Session-Id       | Supported                    | Supported  |
| 50     | Acct-Multi-Session-Id | Supported                    | Supported  |
| 51     | Acct-Link-Count       | Supported                    | Supported  |
| 62     | Port-Limit            | Supported                    | Supported  |
| 124    | Target-Util           | Not Supported                | Supported  |
| 125    | Maximum-Channels      | Supported                    | Supported  |

Following are sample configurations for ML-PPP:

```

/RADIUS
  /Profile
    /Default-ISDN-Users
      Name = Default-ISDN-Users
      Description =
      Attributes/
        Port-Limit = 2
        Target-Util = 70
        Session-Timeout = 70

/RADIUS
  /UserGroups
    /ISDN-Users
      Name = ISDN-Users
      Description = " Users who always use ISDN"
      BaseProfile = Default-ISDN-Users
      Authentication-Script =
      Authorization-Script =

```

The **Port-Limit** attribute controls the number of concurrent sessions a user can have. The **Target-Util** attribute controls the threshold level at which the second B channel should be brought up.

## Dynamic Updates Feature

The Dynamic Updates feature enables changes to server configurations made using **aregcmd** to take effect in the Prime Access Registrar server after issuing the **save** command, eliminating the need for a server **reload** after making changes.

Table 9-6 lists the RADIUS object and its child objects. For each object listed, the **Add** and **Modify or Delete** columns indicate whether a dynamic update occurs after adding, modifying, or deleting an object or attribute. Entries in the **Add** and **Modify or Delete** columns also apply to child objects and child attributes of the objects listed, unless the child object is explicitly listed below the object, such as **/RADIUS/Advanced/Ports** or **/RADIUS/Advanced/Interfaces**.

**Table 9-6** Dynamic Updates Effect on Radius Server Objects

| Object            | Add | Modify or Delete |
|-------------------|-----|------------------|
| Radius            | Yes | Yes              |
| UserLists         | Yes | Yes              |
| UserGroups        | Yes | Yes              |
| Policies          | Yes | Yes              |
| Clients           | Yes | Yes              |
| Vendors           | Yes | Yes              |
| Scripts           | Yes | Yes              |
| Services          | Yes | Yes              |
| SessionManagers   | Yes | No               |
| ResourceManagers  | Yes | No               |
| Profiles          | Yes | Yes              |
| Rules             | Yes | Yes              |
| Translations      | Yes | Yes              |
| TranslationGroups | Yes | Yes              |
| RemoteServers     | Yes | No               |
| Replication       | No  | No               |
| Advanced          | Yes | Yes              |
| SNMP              | No  | No               |
| Ports             | No  | No               |
| Interfaces        | No  | No               |

The Dynamic Updates feature is subject to the following limitations:

- Changes to the Ports or Interfaces objects are not dynamically updated. An **aregcmd reload** command must be issued for these changes to be propagated to the Prime Access Registrar server.
- Changes (modifications and deletions) to existing Session Manager and Resource Manager objects are not dynamically updated. An **aregcmd reload** command must be issued for these changes to be propagated to the Prime Access Registrar server. However, additions of new Session Manager and Resource Manager objects are dynamically updated. Active sessions and allocated resources are preserved in this case.
- Changes to the Prime Access Registrar configuration might not be immediately propagated to the server. Dynamic updates are only carried out in a *safe* environment (that is, when packets are not being processed and when packet processing can be delayed until the changes can be made on the server safely). Dynamic updates will yield to packet processing when appropriate, thus not significantly impacting server performance.
- Changes to SNMP require the Prime Access Registrar server to be restarted (**/etc/init.d/arservagt restart**)

## NAS Monitor

The ability to monitor when a NAS is *down* (really only unreachable from Prime Access Registrar) is provided by **nasmonitor**. This program will repeatedly query a TCP port at the specified IP address until the device (NAS) is reachable. If the NAS is not reachable after a period of time, a warning e-mail is sent; if the NAS is still not reachable after another period of time, a message is sent to Prime Access Registrar to release all sessions associated with that NAS. The port to query, the query frequency, the first time interval, the back-off time interval, and the E-mail address to send to are all configurable (with defaults); the only required parameter is the NAS IP address. This program will work for any device that has a TCP port open; it can either be run by hand, when desired, or put in a **cron** job. See **nasmonitor -h** for details.

**Note**

You must have **telsh** installed in **/usr/local/bin** to use **nasmonitor**. **telsh** is part of the standard Tcl installation that can be downloaded from <http://www.scriptics.com>.

## Automatic Information Collection (arbug)

You can use the script **arbug** to collect information about your Prime Access Registrar server. The results are collected into a tarball that can be e-mailed or **ftped** to Cisco as requested.

**arbug** collects all the relevant information needed to report a problem to Prime Access Registrar support. The goal of the **arbug** script is to make sure all the necessary information is collected.

**Note**

The **arbug** script neither updates nor replaces any system or Prime Access Registrar-related configuration.

This section contains the following topics:

- [Running arbug](#)
- [Files Generated](#)

## Running arbug

To run the **arbug** script, change directory to **/cisco-ar/bin** and enter the following:

```
./arbug
```

The following is a typical sequence.

```
Looking around...
Cluster:
User: admin
Password:
The report /tmp/arbug.10085/arbug.tar is ready to send; you
may want to compress it first using gzip or compress.
hostname user_name bin>
```

## Files Generated

The **arbug** script generates five files that are compressed into a tarball. Table 9-7 provides a summary of the information found in each of the files.

**Table 9-7** Files Generated by *arbug*

| File                       | Description   |
|----------------------------|---|
| <b>car.debug.tar.*</b>     | Machine-specific information including OS type, RAM details, disk space information, swap space information, patch information and open file details. |
| <b>car.config.tar.*</b>    | Prime Access Registrar server configuration, server statistics, database dump by taking the administrator username and password as the input.         |
| <b>car.confini.tar.*</b>   | Information about ODBC <b>.ini</b> files and SNMP configuration   |
| <b>car.core.tar.*</b>      | Core files if any are present   |
| <b>car.logcscrsr.tar.*</b> | Information from scripts directory, certificate directory, license directory  |

## Simultaneous Terminals for Remote Demonstration

Multiple people can view and interact in a single demonstration by using the *share-access* program, a standard GNU release with a special configuration for use with Prime Access Registrar. To run **screen**, a technical support specialist (CSE or DE) will **telnet** to your server and log in as *cisco*. While you run **/opt/CSCOar/bin/share-access** (assuming **/opt/CSCOar** is the Prime Access Registrar path) as *root*, the CSE or DE runs **/opt/CSCOar/bin/share-access -r root**. Now both people (or more) can see what the other types, as well as the results of the commands entered. The special Prime Access Registrar configuration only allows *root* and *cisco* to run **screen**. To end a **share-access** session, type Control-D.

## Support for RADIUS Check Item Attributes

Prime Access Registrar supports RADIUS check item attributes configuration at the user and group levels. You can configure the Prime Access Registrar server to check for attributes that must be present or attributes that must not be present in the Access-Request packet for successful authentication.

When using check item attributes, the Prime Access Registrar server will reject Access-Requests if:

- Any of the configured check item attributes are not present in the Access-Request packet
- Any of the Access-Request packet's check item attribute values do not match with those configured check item attribute values

For remote servers using either LDAP or ODBC, Prime Access Registrar allows for mapping of certain LDAP or ODBC fields to check item attributes. The mapped attributes can be used as check item attributes while processing the Access-Request packets.

When you configure check item attributes at both the user and group levels, the Prime Access Registrar server first checks the attributes of the user level before those of the group level. The Prime Access Registrar server must first authenticate the user's password in the Access-Request before validating the check item attributes.

The Prime Access Registrar server logs details about any rejected Access-Requests as a result of check items processing.

## Configuring Check Items

You use **aregcmd** to configure check item attributes.

### Configuring User Check Items

To configure UserList check item attributes:

- 
- Step 1** Log into the Prime Access Registrar server, and use **aregcmd** to navigate to **//localhost/Radius/UserLists/default/bob**.

```
[ //localhost/Radius/UserLists/Default/bob ]
Name = bob
Description =
Password = <encrypted>
AllowAnonymousPassword = FALSE
Enabled = TRUE
Group~ = PPP-users
BaseProfile~ =
AuthenticationScript~ =
AuthorizationScript~ =
UserDefined1 =
Attributes/
CheckItems/
```

- Step 2** Change directory to CheckItems.

#### **cd CheckItems**

```
[ //localhost/Radius/UserLists/Default/bob/CheckItems ]
```

- Step 3** Use **set** to add any attributes to be used as check items.

```
set calling-Station-Id 4085551212
```

```
save
```

---

### Configuring Usergroup Check Items

To configure UserGroups check item attributes:

- 
- Step 1** Log into the Prime Access Registrar server, and use **aregcmd** to navigate to **//localhost/Radius/UserGroups/Default**.

#### **cd /Radius/UserGroups/Default**

```
[ //localhost/Radius/UserGroups/Default ]
Name = Default
Description = "Users who sometimes connect using PPP and sometimes connect "
BaseProfile~ =
AuthenticationScript~ =
AuthorizationScript~ = AuthorizeService
Attributes/
CheckItems/
```

- Step 2** Change directory to CheckItems.

**cd CheckItems**

```
[ //localhost/Radius/UserGroups/Default/CheckItems ]
```

**Step 3** Use set to add any attributes to be used as check items.

```
set NAS-IP-Address 10.10.10.10
```

```
save
```

---

## User-Specific Attributes

The Prime Access Registrar server supports user-specific attributes which enables the Prime Access Registrar server to return attributes on a per-user or per-group basis without having to use profiles.

The Prime Access Registrar server includes a property called HiddenAttributes to the User and UserGroup object. The HiddenAttributes property contains a concatenation of all user-level reply attributes. The HiddenAttributes property is not displayed, nor can the value be set or unset using the command-line interface.

The order of application of attributes is as follows:

1. UserGroup Base Profile
2. UserGroup Attributes
3. User Base Profile
4. User Attributes

The value of the HiddenAttributes property is used dynamically to construct and populate a virtual *attributes* directory in the User object. All values from the Attributes directory will go into the HiddenAttributes property. This occurs transparently when the administrator issues a save command.

## Packet of Disconnect

Prime Access Registrar supports the Packet of Disconnect (POD) feature that enables the Prime Access Registrar server to send disconnect requests (PODs) to a NAS so that all the session information and the resources associated with the user sessions can be released. Prime Access Registrar can also determine when to trigger and send the POD.

For example, when a PDSN handoff occurs during a mobile session, the new PDSN sends out a new access-request packet to Prime Access Registrar for the same user. Prime Access Registrar should detect this handoff by the change in NAS-Identifier in the new request and trigger sending a POD to the old PDSN if it supports POD. Prime Access Registrar also provides an option for administrator to initiate sending POD requests through the command-line interface (CLI) for any user session. Prime Access Registrar forwards POD requests from external servers to the destination NAS.

This section contains the following topics:

- [Configuring Packet of Disconnect](#)
- [Proxying POD Requests from External Servers](#)

- [CLI Options for POD](#)

## Configuring Packet of Disconnect

This section describes how to configure the POD feature in the following:

- [Configuring the Client Object](#)
- [Configuring a Resource Manager for POD](#)

### Configuring the Client Object

You should enable POD for each client object that might want to send disconnect requests to those clients. You enable POD in a client object using the `EnableDynamicAuthorization` property. This property is set to `FALSE` by default when you create a client object. The following example shows the default configuration for a new client object, `NAS1`.

```
[ //localhost/Radius/Clients/NAS1 ]
  Name = nas1
  Description =
  IPAddress =
  SharedSecret =
  Type = NAS
  Vendor =
  IncomingScript~ =
  OutgoingScript~ =
  EnableDynamicAuthorization = FALSE
```

If the Prime Access Registrar server might send a POD to this client, set the `EnableDynamicAuthorization` property to `TRUE`. When you set this property to `TRUE`, the Prime Access Registrar server creates a `DynamicAuthorizationServer` subdirectory under the client object. The following example shows a newly created `DynamicAuthorizationServer` subdirectory:

```
[ //localhost/Radius/Clients/NAS1/DyanamicAuthorizationServer ]
  Port = 3799
  DynamicAuthSharedSecret =
  InitialTimeout = 5000
  MaxTries = 3
  PODAttributeGroup =
  COAAttributeGroup =
```

The default port is 3799. You can change the port, if desired.

The property `DynamicAuthSharedSecret` is initially set to the same as value as the client's `SharedSecret` property when you set `EnableDynamicAuthorization` to `TRUE`. You can chose to configure a different secret for POD in this subdirectory.

The `InitialTimeout` property represents the number of milliseconds used as a timeout for the first attempt to send a POD packet to a remote server. For each successive retry on the same packet, the previous timeout value used is doubled. You must specify a number greater than zero, and the default value is 5000 (or 5 seconds).

The `MaxTries` property represents the number of times to send a proxy request to a remote server before deciding the server is offline. You must specify a number greater than zero, and the default is 3.

The `PODAttributeGroup` property points to a group of attributes to be included in a disconnect-request packet sent to this client.

You can create and configure the PODAttributeGroup in the **/Radius/Advanced/AttributeGroups/** directory. The default group contains commonly used POD attributes NAS-Port and Acct-Session-Id. The COAAttributeGroup property is used with the Change of Authorization (CoA) feature, also known as hot-lining.

## Configuring a Resource Manager for POD

Prime Access Registrar provides a resource manager type called *session-cache*. When you set a resource manager to session-cache, the resource manager's configuration contains a subdirectory called **AttributesToBeCached**. The following is an example Resource Manager set to type session-cache:

```
[ //localhost/Radius/ResourceManagers/PODresourceMgr ]
  Name = PODresourceMgr
  Description =
  Type = session-cache
  OverwriteAttributes = FALSE
  AttributesToBeCached/
  QueryMappings/
```

The attributes you configure under the **AttributesToBeCached** directory are cached in the session record during session management. The cached attributes are then sent in the disconnect-request for this session.

The OverwriteAttributes property indicates whether to overwrite the existing attributes if there are any in the session record. Since this resource manager can be invoked during Access-Request as well as Accounting-Start processing, the OverwriteAttributes can be used to control if the attributes cached during Access-Request processing can be overwritten with the attributes available during Accounting-Start processing.

The following is an example of a typical session-cache resource manager:

```
[ //localhost/Radius/ResourceManagers/RM-New ]
  Name = RM-New
  Description =
  Type = session-cache
  OverwriteAttributes = TRUE
  AttributesToBeCached/
    1. Framed-IP-Address
    2. CDMA-Correlation-ID
  QueryMappings/
```

The attributes used in the example can be added as an indexed list using **add** or **set** commands (in any order).

## Proxying POD Requests from External Servers

Prime Access Registrar can also proxy the disconnect requests received from external servers. To make Prime Access Registrar listen for external POD requests, the ListenForDynamicAuthorizationRequests property under **/Radius/Advanced** should be set to TRUE. The default value for this is FALSE. The default POD listening port is 3799. However this can be changed by configuring a new port of type *pod* under **/Radius/Advanced/Ports** and setting the new port number accordingly.

For security reasons, the source of a POD request should be configured as a remote server in Prime Access Registrar and the remote server should be configured to accept PODs. Set the property AcceptDynamicAuthorizationRequests to TRUE to do this. The default for this is FALSE. POD requests from unauthorized sources are silently discarded.

## CLI Options for POD

Prime Access Registrar has options for the **query-sessions** and **release-sessions** CLI commands that enable querying or releasing sessions based on the session's age. Another option enables querying or releasing sessions based on any valid RADIUS attribute available in the user's session record. This section contains the following topics:

- [query-sessions](#)
- [release-sessions](#)

### query-sessions

The syntax for using **query-sessions** *with-Age* option is the following:

```
query-sessions <path> with-Age <value>
```

Where <path> is the path to the server, session-manager or resource manager and <value> is the minimum age of the session specified in minutes or hours with options M, Minutes, H or Hours. This command returns all sessions that are older than the given age value.

The syntax for using **query-sessions** *with-Attribute* option is the following:

```
query-sessions <path> with-Attribute <name> <value>
```

Where <name> is the RADIUS attribute name and <value> is the value of the attribute to be matched. This command returns the sessions where a session record contains and matches the attribute value specified in <value> field.

### release-sessions

The syntax for using **release-sessions** *with-Age* option is:

```
release-sessions <path> with-Age <value>
```

Where, <path> is the path to the server, session-manager or resource manager and <value> is the minimum age of the session specified in minutes or hours with options M for Minutes, H for Hours. This command returns all sessions that are older than the given age value.

The syntax for using **release-sessions** *with-Attribute* option is:

```
release-sessions <path> with-Attribute <name> <value>
```

Where, <name> is the RADIUS attribute name and <value> is the value of the attribute to be matched. This command returns the sessions where a session record contains and matches the attribute value specified in <value> field.

A new option is also available for **release-sessions** command to enable an administrator to trigger sending a POD for a user after the session is released.

```
release-sessions <path> with-<type> <value> [send-pod]
```

Where, <path> is the path to the server, Session Manager, or Resource Manager and <type> is one of the following: NAS, User, IP-Address ID, or Age. The **release-sessions** command with an optional [send-pod] at the end results in Prime Access Registrar sending a POD request. The PoD requests are directed to port number configured in /radius/clients/<client name>/DynamicAuthorizationServer/port. By default it is set to 3799. To configure udp xxx, set the port value as:

```
/radius/clients/<client name>/DynamicAuthorizationServer/port = xxx
```

## Configuring Change of Authorization Requests

Prime Access Registrar supports Change of Authorization (CoA) requests as defined in Internet RFC 3576 that provides a way to change authorization status of users already logged on to the network. The CoA feature, also known as hot-lining, provides a wireless operator the ability to efficiently address issues with users that might otherwise be unauthorized to access packet data services. When a problem occurs that causes a user to be unauthorized to use the packet data service, a wireless operator can use the CoA feature to resolve the problem and return the user's packet data services.

When a user is hot-lined, their packet data service is redirected to a hot-line application that notifies the user of issues that might be blocking their access to normal packet data services. Hot-lining provides users with a way to address the issues blocking their access, such as billing issues, a prepaid account that has been depleted, or an expired credit card.

The CoA feature provides an option to the wireless operator administrator to send CoA packets to the client device when a user needs to be hot-lined. When to send a CoA request to a user depends on the wireless operator's site-specific policies.

## Configuring the Client Object

You should enable CoA for each client object that might want to send CoA requests to those clients. You enable CoA in a client object using the EnableDynamicAuthorization property. This property is set to FALSE by default when you create a client object. The following example shows the default configuration for a new client object, NAS1.

```
[ //localhost/Radius/Clients/NAS1 ]
  Name = nas1
  Description =
  IPAddress =
  SharedSecret =
  Type = NAS
  Vendor =
  IncomingScript~ =
  OutgoingScript~ =
  EnableDynamicAuthorization = FALSE
```

If the Prime Access Registrar server might send a CoA request to this client, set the EnableDynamicAuthorization property to TRUE. When you set this property to TRUE, the Prime Access Registrar server creates a DynamicAuthorizationServer subdirectory under the client object. The following example shows a newly created DynamicAuthorizationServer subdirectory:

```
[ //localhost/Radius/Clients/NAS1/COA ]
  Port = 3799
  DynamicAuthSharedSecret =
  InitialTimeout = 5000
  MaxTries = 3
  PODAttributeGroup =
  COAAttributeGroup =
```

The default port is 3799. You can change the port, if desired.

The property `DynamicAuthSharedSecret` is initially set to the same as value as the client's `SharedSecret` property when you set `EnableDynamicAuthorization` to `TRUE`. You can chose to configure a different secret for CoA in this subdirectory.

The `InitialTimeout` property represents the number of milliseconds used as a timeout for the first attempt to send a CoA packet to a remote server. For each successive retry on the same packet, the previous timeout value used is doubled. You must specify a number greater than zero, and the default value is 5000 (or 5 seconds).

The `MaxTries` property represents the number of times to send a proxy request to a remote server before deciding the server is offline. You must specify a number greater than zero, and the default is 3.

The `COAAttributeGroup` property points to a group of attributes to be included in a CoA request packet sent to this client.

You can create and configure the `COAAttributeGroup` in the `/Radius/Advanced/AttributeGroups/` directory. The default group is not set to any value by default. When an attribute group is configured, the Prime Access Registrar server includes the attributes in this group in a CoA request. The values for these attributes are fetched from the user's session record.

The CoA attribute group configuration can be used with a session-cache Resource Manager. For example, any new attributes that are to be sent in a CoA request can be configured for caching by the session-cache Resource Manager so they will be available in the session record when it is to be sent in the CoA request.

The CoA request might also contain AV pairs from the optional profile name in the `query-session` CLI command used to send the CoA request. In a 3GPP2 scenario, a profile containing the `Filter-Id` attribute set to a value "Hot-Line Active" can be included when a user is to be hot-lined. This can be used as a hot-line profile possibly containing other attributes as desired by the wireless operator. Another profile might be defined containing the `Filter-Id` attribute with the value "Hot-Line Normal." This profile can be used with the `query-session` CLI command to bring the user back to normal.

The CoA request packet sent by the Prime Access Registrar server conforms to internet RFC 3756. In response to a CoA request initiated by the Prime Access Registrar server, the client should respond with a `COA-ACK` if it is able to hot-line the user based on credentials available in the CoA request. If the client is unable to hot-line the user for any reason, the client can include an error-cause attribute with the appropriate reason in a `COA-NAK` packet.

The Prime Access Registrar server logs all CoA responses. If the Prime Access Registrar server does not receive a response to a CoA request within the timeout period, it will retransmit for the configured number of retries, then logs an error if no response is received.

The Prime Access Registrar server forwards proxied CoA requests sent by external servers to the destination NAS. The CoA requests are proxied based on the `NAS-IP-Address` in the incoming request. The proxied CoA requests from external servers are forwarded to the destination NAS only if the source IP address is configured to accept dynamic authorization requests. The responses received from the NAS (either `COA-ACK` or `COA-NAK`) are forwarded back to the source where the Prime Access Registrar server received the original proxy request.

## Dynamic DNS

Prime Access Registrar supports the Dynamic DNS protocol providing the ability to update DNS servers. The dynamic DNS updates contain the hostname/IP Address mapping for sessions managed by Prime Access Registrar.

You enable dynamic DNS updates by creating and configuring new Resource Managers and new Remote Servers, both of type *dynamic-dns*. The dynamic-dns Resource Managers specify which zones to use for the forward and reverse zones and which Remote Servers to use for those zones. The dynamic-dns Remote Servers specify how to access the DNS Servers.

This section contains the following topics:

- [Configuring Dynamic DNS](#)
- [Testing Dynamic DNS with radclient](#)

## Configuring Dynamic DNS

Before you configure Prime Access Registrar you need to gather information about your DNS environment. For a given Resource Manager you must decide which forward zone you will be updating for sessions the resource manager will manage. Given that forward zone, you must determine the IP address of the primary DNS server for that zone. If the dynamic DNS updates will be protected with TSIG keys, you must find out the name and the base64 encoded value of the secret for the TSIG key. If the resource manager should also update the reverse zone (ip address to host mapping) for sessions, you will also need to determine the same information about the primary DNS server for the reverse zone (IP address and TSIG key).

If using TSIG keys, use **aregcmd** to create and configure the keys. You should set the key in the Remote Server or the Resource Manager, but not both. Set the key on the Remote Server if you want to use the same key for all of the zones accessed through that Remote Server. Otherwise, set the key on the Resource Manager. That key will be used only for the zone specified in the Resource Manager.

### Configuring the Dynamic DNS

To configure the dynamic-dns remote server:

---

**Step 1** Launch **aregcmd**.

**Step 2** Create the dynamic-dns TSIG Keys:

```
cd /Radius/Advanced/DDNS/TSIGKeys
add foo.com
```

This example named the TSIG Key, **foo.com**, which is related to name of the example DNS server we use. You should choose a name for TSIG keys that reflects the DDNS client-server pair (for example, **foo.bar** if the client is **foo** and the server is **bar**), but you should use the name of the TSIG Key as defined in the DNS server.

**Step 3** Configure the TSIG Key:

```
cd foo.com
set Secret <base64-encoded string>
```

The Secret should be set to the same base64-encoded string as defined in the DNS server. If there is a second TSIG Key for the primary server of the reverse zone, follow these steps to add it, too.

**Step 4** Use **aregcmd** to create and configure one or more dynamic-dns Remote Servers.

**Step 5** Create the dynamic-dns remote server for the forward zone:

```
cd /Radius/RemoteServers
```

**add ddns**

This example named the remote server *ddns* which is related to the remote server type. You can use any valid name for your remote server.

**Step 6** Configure the dynamic-dns remote server:

```
cd ddns
```

```
set Protocol dynamic-dns
```

```
set IPAddress 10.10.10.1 (ip address of primary dns server for zone)
```

```
set ForwardZoneTSIGKey foo.com
```

```
set ReverseZoneTSIGKey foo.com
```

If the reverse zone will be updated and if the primary server for the reverse zone is different than the primary server for the forward zone, you will need to add another Remote Server. Follow the previous two steps to do so. Note that the IP Address and the TSIG Key will be different.

You can now use **aregcmd** to create and configure a resource manager of type dynamic-dns.

**Step 7** Create the dynamic-dns resource manager:

```
cd /Radius/ResourceManagers
```

```
add ddns
```

This example named the service *ddns* which is related to the resource manager type but you can use any valid name for your resource manager.

**Step 8** Configure the dynamic-dns resource manager.

```
cd ddns
```

```
set Type dynamic-dns
```

```
set ForwardZone foo.com
```

```
set ForwardZoneServer DDNS
```

Finally, reference the new resource manager from a session manager. Assuming that the example configuration was installed, the following step will accomplish this. If you have a different session manager defined you can add it there if that is appropriate.

**Step 9** Reference the resource manager from a session manager:

```
cd /Radius/SessionManagers/session-mgr-1/ResourceManagers
```

```
set 5 DDNS
```

**Note**

The Property AllowAccountingStartToCreateSession must be set to TRUE for dynamic DNS to work.

**Step 10** Save the changes you have made.

## Testing Dynamic DNS with radclient

After the Resource Manager has been defined it must be referenced from the appropriate Session Manager. You can use **radclient** to confirm that dynamic DNS has been properly configured and is operational.

### Testing the Dynamic DNS using Radclient

To test Dynamic DNS using radclient:

---

**Step 1** Launch **aregcmd** and set the trace to level 4.

```
aregcmd
```

Login to the Prime Access Registrar server as an administrative user.

```
trace 4
```

**Step 2** Launch **radclient**.

```
cd /opt/CSCOAr/bin
```

```
radclient
```

**Step 3** Create an Accounting-Start packet

```
acct_request Start username
```

Example:

```
set p [ acct_request Start bob ]
```

**Step 4** Add a Framed-IP-Address attribute to the Accounting-Start packet

**Step 5** Send the Accounting-Start packet

```
$p send
```

**Step 6** Check the **aregcmd** trace log and the dns server to verify that the host entry was updated in both the forward and reverse zones.

---

## Dynamic Service Authorization Feature

Typically, Prime Access Registrar does not allow sending another Access-Request to the remote server after the user is connected to the LDAP servers for user authentication. The Dynamic Service Authorization feature allows you to access external databases such as LDAP and Oracle first to know which remote servers authenticated services need to be relayed. This feature enables Prime Access Registrar to determine whether to send access-accept back to the client or to send another access-request to the remote server such as LDAP and Oracle. Prime Access Registrar is able to perform this activity multiple times in a single access-request.

## Configuring Dynamic Service Authorization Feature

Configuring the dynamic service authorization involves:

- [Setting Up the Environment Variable](#)
- [Configuring the Script for the Dynamic Service Authorization](#)

## Setting Up the Environment Variable

Before configuring the dynamic service authorization feature, you must set the following three environment variables in Prime Access Registrar:

- **Re-Authentication-Service**

When the Re-Authentication-Service is set, the server directs the request to the specified reauthentication service for processing.

- **Re-Authorization-Service**

When the Re-Authorization-Service is set, the server directs the request to the specified reauthorization service for processing.

- **Re-Accounting-Service**

When the Re-Accounting-Service is set, the server directs the request to the specified reaccounting service for processing.

You can set the environmental variable by using scripts. See for more information.



### Note

When using the same service for reauthentication and reauthorization, a loop can occur in these services. The loop count, by default is 10. You can change the loop count using the **Dynamic-Service-Loop-Limit** environment variable.

Following is a sample procedure for setting the environment variable:

```
proc dynamic-service { request response environ } {
  $environ put Re-Authentication-Service "local-users"
  $environ put Re-Authorization-Service "local-users"
}
```

You can append this procedure by copying it into **tclscript.tcl** that is located in **/opt/CSCOar/scripts/radius/tcl directory**, or to the location that you chose when you installed Prime Access Registrar. You can also use this procedure as a separate script file and configure the script accordingly. See for more information on configuring the TCL script.

### Configuring the Script for the Dynamic Service Authorization

To configure the script for the dynamic service authorization:

- 
- Step 1** Launch **aregcmd**.  
**aregcmd**
  - Step 2** Change directory to **/Radius/Scripts**.  
**cd /Radius/Scripts**
  - Step 3** Enter **dynamic-service**.
  - Step 4** Change the directory to **dynamic-service**.  
**cd dynamic-service**

You get the following output:

```
[ //localhost/Radius/Scripts/dynamic-service ]
Name = dynamic-service
Description =
Language =
```

- Step 5** Set the Language property to TCL.

**Set Language TCL**

**Step 6** Set the filename property to **tclscript.tcl**.

**Set Filename tclscript.tcl**

**Step 7** Set the EntryPoint property to **dynamicservice**.

**Set EntryPoint dynamicservice**

The following is an example of the script configuration:

```
cd /Radius
set IncomingScript dynamicservice
[ //localhost/Radius ]
  IncomingScript~ = dynamicservice
  DefaultAuthenticationService~ = local-users
  DefaultAuthorizationService~ = local-users
```

**Step 8** Enter **Save** to save the configuration.

The following shows a sample trace:

```
10/30/2013 12:32:02.258: P577: Packet received from 127.0.0.1
10/30/2013 12:32:02.259: P577: Packet successfully added
10/30/2013 12:32:02.259: P577: Trace of Access-Request packet
10/30/2013 12:32:02.259: P577:   identifier = 9
10/30/2013 12:32:02.259: P577:   length = 61
10/30/2013 12:32:02.259: P577:   reqauth =
b6:89:41:52:6e:d4:86:37:4a:aa:9b:27:1f:74:ff:05
10/30/2013 12:32:02.259: P577:   User-Name = bob
10/30/2013 12:32:02.259: P577:   User-Password =
2b:4a:f0:c8:95:f1:ad:e5:52:d4:83:0f:45:2b:2b:70
10/30/2013 12:32:02.259: P577:   NAS-Port = 2
10/30/2013 12:32:02.260: P577:   NAS-Identifier = localhost
10/30/2013 12:32:02.260: P577: Running Server's IncomingScript: dynamicservice
10/30/2013 12:32:02.261: P577:   Tcl: environ put Re-Authentication-Service local-users
-> OK
10/30/2013 12:32:02.261: P577:   Tcl: environ put Re-Authorization-Service local-users
-> OK
10/30/2013 12:32:02.261: P577: Using Client: localhost
10/30/2013 12:32:02.262: P577: Using NAS: localhost (127.0.0.1)
10/30/2013 12:32:02.262: P577: Request is directly from a NAS: TRUE
10/30/2013 12:32:02.262: P577: Authenticating and Authorizing with Service local-users
10/30/2013 12:32:02.262: P577: Getting User bob's UserRecord from UserList Default
10/30/2013 12:32:02.263: P577: user list user bob's password matches
10/30/2013 12:32:02.263: P577: Processing UserGroup PPP-users's check items
10/30/2013 12:32:02.263: P577: User bob is part of UserGroup PPP-users
10/30/2013 12:32:02.263: P577: Merging UserGroup PPP-users's BaseProfiles into response
dictionary
10/30/2013 12:32:02.264: P577: Merging BaseProfile default-PPP-users into response
dictionary
10/30/2013 12:32:02.264: P577: Merging attributes into the Response Dictionary:
10/30/2013 12:32:02.264: P577:   Adding attribute Service-Type, value = Framed
10/30/2013 12:32:02.264: P577:   Adding attribute Framed-Protocol, value = PPP
10/30/2013 12:32:02.264: P577:   Adding attribute Framed-Routing, value = None
10/30/2013 12:32:02.264: P577:   Adding attribute Framed-MTU, value = 1500
10/30/2013 12:32:02.264: P577:   Adding attribute Framed-Compression, value = VJ TCP/IP
header compression
10/30/2013 12:32:02.264: P577:   Adding attribute Ascend-Idle-Limit, value = 1800
10/30/2013 12:32:02.265: P577: Merging UserGroup PPP-users's Attributes into response
Dictionary
10/30/2013 12:32:02.265: P577: Merging attributes into the Response Dictionary:
10/30/2013 12:32:02.265: P577: Authenticating and Authorizing with Service local-users
10/30/2013 12:32:02.265: P577: Getting User bob's UserRecord from UserList Default
10/30/2013 12:32:02.266: P577: user list user bob's password matches
```

```

10/30/2013 12:32:02.266: P577: Processing UserGroup PPP-users's check items
10/30/2013 12:32:02.266: P577: User bob is part of UserGroup PPP-users
10/30/2013 12:32:02.266: P577: Merging UserGroup PPP-users's BaseProfiles into response
dictionary
10/30/2013 12:32:02.266: P577: Merging BaseProfile default-PPP-users into response
dictionary
10/30/2013 12:32:02.266: P577: Merging attributes into the Response Dictionary:
10/30/2013 12:32:02.266: P577:   Replacing attribute Service-Type, new value = Framed
10/30/2013 12:32:02.267: P577:   Replacing attribute Framed-Protocol, new value = PPP
10/30/2013 12:32:02.267: P577:   Replacing attribute Framed-Routing, new value = None
10/30/2013 12:32:02.267: P577:   Replacing attribute Framed-MTU, new value = 1500

```

## Remote Session Management

Prime Access Registrar sessions can also be stored on a remote database. This improves the overall scalability of the number of sessions that Prime Access Registrar can simultaneously handle. The remote session manager internally uses two ODBC remote servers, Internal-ODBC-Read-Server and Internal-ODBC-Write-Server. Configurations pertaining to these internal remoteservers can be done under **/Radius/Advanced/RemoteODBCSessionServer**

For more information on how to configure the Remote ODBC Session Server, refer to .



### Note

Ensure that the length of fields such as Username, Session/Resource Manager name Session-Key, Query-Key and so on are limited to the value specified in the [Table 9-8](#) while it is configured. Although the field length of entire session record is 3KB it is limited to 2KB. This is practically sufficient to hold all the session parameters as well as the cached attributes (if any).

**Table 9-8** Schema Details

| Field           | Type           |
|-----------------|----------------|
| ID              | NUMBER(10)     |
| SESSION_KEY     | VARCHAR2(20)   |
| NAME            | VARCHAR2(20)   |
| PER_USER_RM     | VARCHAR2(20)   |
| PER_GROUP_RM    | VARCHAR2(20)   |
| IP_RM           | VARCHAR2(20)   |
| IP              | VARCHAR2(20)   |
| SESSION_MANAGER | VARCHAR2(20)   |
| AC              | NUMBER(10)     |
| NAS             | VARCHAR2(20)   |
| CACHE_RM        | VARCHAR2(20)   |
| Q_VALUE         | VARCHAR2(20)   |
| TS              | NUMBER(15)     |
| SESSION_RECORD  | VARCHAR2(3072) |

**Note**

Remote session manager will work only with Oracle database.

**Note**

In remote-session-manager, query-session with the 'with-age' option will not work.

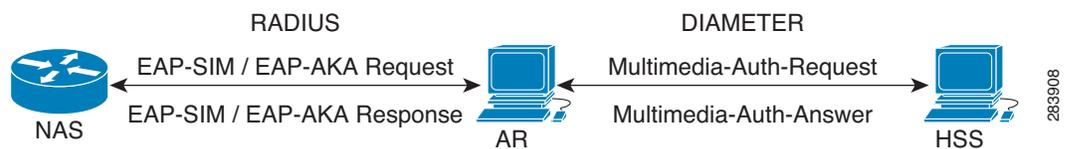
## Wx Interface Support for SubscriberDB Lookup

Prime Access Registrar supports Diameter Wx interface to fetch the authentication vectors from HSS required for EAP-SIM/EAP-AKA authentication.

The EAP-SIM and EAP-AKA authentication service is extended to generate a Diameter message Multimedia-Authentication-Request (MAR), with the subscriber identity (IMSI), to the HSS when it requires the authentication vectors. The HSS sends a Diameter Multimedia-Authentication-Answer (MAA) back containing the number of triplets/quintuplets.

The PreRequestTranslationScript, PostRequestTranslationScript, PreResponseTranslationScript, and PostResponseTranslationScript are the available scripting points to modify the RADIUS and Diameter packets while sending and receiving the packets to or from the HSS. For more information, see [Table 5-1](#) for EAP-AKA and for EAP-SIM details.

**Figure 9-1** Wx Interface Support for SubscriberDB lookup



For more information on Wx interface, see the [3GPP TS 29.124](#) and [TS 29.229](#) specifications.

## Configuration Examples

The following shows an example configuration for EAP-AKA:

```
[ //localhost/Radius/Services/eap-aka-wx ]
  Name = eap-aka-wx
  Description =
  Type = eap-aka
  AlwaysRequestIdentity = False
  EnableIdentityPrivacy = False
  PseudonymSecret = <encrypted>
  PseudonymRenewtime = "24 Hours"
  PseudonymLifetime = Forever
  Generate3GPPCompliantPseudonym = False
  EnableReauthentication = False
  MaximumReauthentications = 16
  ReauthenticationTimeout = 3600
```

```

ReauthenticationRealm =
AuthenticationTimeout = 120
QuintetGenerationScript~ =
UseProtectedResults = False
SendReAuthIDInAccept = False
SubscriberDBLookup = Diameter
DestinationRealm = mpc.com
PreRequestTranslationScript~ =
PostRequestTranslationScript~ =
PreResponseTranslationScript~ =
PostResponseTranslationScript~ =

```

The following shows an example configuration for EAP-SIM:

```

[ //localhost/Radius/Services/eap-sim-wx ]
Name = eap-sim-wx
Description =
Type = eap-sim
NumberOfTriplets = 2
UseSimDemoTriplets = False
AlwaysRequestIdentity = False
EnableIdentityPrivacy = False
PseudonymSecret = <encrypted>
PseudonymRenewtime = "24 Hours"
PseudonymLifetime = Forever
Generate3GPPCompliantPseudonym = False
EnableReauthentication = False
MaximumReauthentications = 16
ReauthenticationTimeout = 3600
ReauthenticationRealm =
TripletCacheTimeout = 120
AuthenticationTimeout = 120
UseProtectedResults = False
SendReAuthIDInAccept = False
SubscriberDBLookup = Diameter
DestinationRealm = hss.com
PreRequestTranslationScript~ =
PostRequestTranslationScript~ =
PreResponseTranslationScript~ =
PostResponseTranslationScript~ =

```

## Smart Grid Solution Management

Prime Access Registrar provides identity and access management for the smart grid solutions on IPv6 (and IPv4) networks. This is achieved using the Elliptic Curve Cryptographic (ECC) based certificate validation and SNMP support for TACACS+.

For EAP services, in addition to RSA certificates, Prime Access Registrar supports verification of ECC certificates. ECC uses elliptic curves to encrypt data when creating keys which enables you to create shorter and stronger keys for better efficiency. This is achieved using the Cisco SSL library APIs.

TACACS+ supports ASCII,PAP, and CHAP Authentication type, login and enable services, and LDAP, OCI, and ODBC services in addition to Local service.

The client certificate files and RSA or ECC key file are available in `/cisco-ar/pki` as **client-cert.pem** and **client-key.pem** respectively. Both the files must be in “.PEM” format, since the certificate validation is done based on the extension of the files.

ECC certificate validation is used in the following authentication methods:

- [EAP-FAST](#)
- [EAP-Transport Level Security \(TLS\)](#)
- [EAP-TTLS](#)
- [Protected EAP](#)

## Lawful Interception (LI) Support in Prime Access Registrar

Lawful Interception (LI) is a requirement placed upon service providers to provide legally sanctioned official access to private communications. With the existing Public Switched Telephone Network (PSTN), LI is performed by applying a physical tap on the telephone line of the target in response to a warrant from a Law Enforcement Agency (LEA). However, Voice over IP (VoIP) technology has enabled the mobility of the end-user, so it is no longer possible to guarantee the interception of calls based on tapping a physical line.

When a Law Interception Server (LIS) of the LEA requests the LI server to start monitoring a particular target, LI server sends the corresponding request to the Prime Access Registrar server. XML schema definition files are shared between Prime Access Registrar and Mediation Partner device for request and response messages. A local web service, which runs on the Prime Access Registrar server listens to the messages from the LI server.

Prime Access Registrar provides support for Intercept Access Point (IAP) for receiving the intercept/monitoring request for the subscriber whose “Access Associated” Communications Identifying Information (AA CmII) is to be intercepted and delivered to the LIS.

[Table 9-9](#) provides the list of supported RADIUS and Diameter intercept requests from the LIS.

**Table 9-9 Intercept Requests Supported**

| Intercept Request (RADIUS) | Intercept Request (Diameter) | Purpose                             |
|----------------------------|------------------------------|-------------------------------------|
| ProvisionTargetRequest     | DiaProvisionTargetRequest    | To start monitoring the target user |
| DeprovisionTargetRequest   | DiaDeProvisionTargetRequest  | To stop monitoring the target user  |

**Table 9-9 Intercept Requests Supported**

| Intercept Request (RADIUS) | Intercept Request (Diameter) | Purpose  |
|----------------------------|------------------------------|--|
| LinkUpdateRequest          | DiaLinkUpdateRequest         | To query the target user in the monitored list           |
| ListTargetRequest          | DiaListTargetRequest         | To list all the users that are currently being monitored |

**Initiating Monitoring Process**

When the ProvisionTarget/DiaProvisionTarget request is received from the LIS, Prime Access Registrar adds the respective user in the monitoring list and starts monitoring the user events.

Table 9-10 lists the events of the target user that are reported to LIS:

**Table 9-10 Targeted User Events**

| Events  | Attributes (RADIUS)  | Attributes (Diameter)  |
|---|--|--|
| Access Attempt (for RADIUS) /<br>DiameterAccess Attempt (for<br>Diameter) | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• AccessMethod (C)</li> <li>• NetworkAccessNodeIdentity (C)</li> <li>• ProtocolSignal (O)</li> </ul>  | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• OriginHost (C)</li> <li>• AuthRequestType (C)</li> <li>• SessionIdentity (C)</li> <li>• AuthApplID (C)</li> <li>• ProtocolSignal (O)</li> <li>• OriginRealm (C)</li> <li>• TargetNetwork (O)</li> </ul>                           |
| Access-Accept (for RADIUS) /<br>DiameterAccess-Accept (for Diameter)      | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• AccessMethod (C)</li> <li>• NetworkAccessNodeIdentity (C)</li> <li>• IPaddress (C)</li> <li>• AccessSessionIdentity (M)</li> <li>• AccessSessionCharacteristics (C)</li> <li>• Locationinformation (C)</li> <li>• ProtocolSignal (O)</li> </ul> | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• OriginHost (C)</li> <li>• AuthRequestType (C)</li> <li>• SessionIdentity (C)</li> <li>• AuthApplID (C)</li> <li>• ProtocolSignal (O)</li> <li>• OriginRealm (C)</li> <li>• TargetNetwork (O)</li> <li>• ResultCode (C)</li> </ul> |

**Table 9-10 Targeted User Events (continued)**

| Events  | Attributes (RADIUS)  | Attributes (Diameter)  |
|---|--|--|
| Access-Failed (for RADIUS) /<br>DiameterAccess-Failed (for Diameter)                  | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• IPAddress (C)</li> <li>• ReasonForTermination (C)</li> <li>• ProtocolSignal (O)</li> </ul>  | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• OriginHost (C)</li> <li>• AuthRequestType (C)</li> <li>• SessionIdentity (C)</li> <li>• AuthApplID (C)</li> <li>• ProtocolSignal (O)</li> <li>• OriginRealm (C)</li> <li>• TargetNetwork (O)</li> <li>• ResultCode (C)</li> <li>• ReasonForTermination (C)</li> </ul> |
| Access-Session-Start (for RADIUS) /<br>DiameterAccess-Session-Start (for<br>Diameter) | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• AccessSessionIdentity (M)</li> <li>• IPAddress (C)</li> <li>• ProtocolSignal (O)</li> </ul> | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• OriginHost (C)</li> <li>• AuthApplID (C)</li> <li>• SessionIdentity (M)</li> <li>• AuthRecNo (C)</li> <li>• ProtocolSignal (O)</li> <li>• OriginRealm (C)</li> <li>• TargetNetwork (O)</li> </ul>   |

Table 9-10 Targeted User Events (continued)

| Events  | Attributes (RADIUS)  | Attributes (Diameter)  |
|---|--|--|
| Access-Session-End (for RADIUS) /<br>DiameterAccess-Session-End (for<br>Diameter) | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• AccessSessionIdentity (M)</li> <li>• IPAddress (C)</li> <li>• ReasonforTermination (C)</li> <li>• ProtocolSignal (O)</li> </ul> | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• OriginHost (C)</li> <li>• AuthApplID (C)</li> <li>• SessionIdentity (M)</li> <li>• AuthRecNo (C)</li> <li>• ProtocolSignal (O)</li> <li>• OriginRealm (C)</li> <li>• TargetNetwork (O)</li> <li>• ReasonForTermination (C)</li> </ul>                                 |
| Access-Rejected (for RADIUS) /<br>DiameterAccess-Rejected (for<br>Diameter)       | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• IPAddress (C)</li> <li>• ReasonforTermination (C)</li> <li>• ProtocolSignal (O)</li> </ul>                                      | <ul style="list-style-type: none"> <li>• CaseIdentity (M)</li> <li>• IAPSystemIdentity (M)</li> <li>• TimeStamp (M)</li> <li>• SubscriberIdentity (M)</li> <li>• OriginHost (C)</li> <li>• AuthRequestType (C)</li> <li>• SessionIdentity (C)</li> <li>• AuthApplID (C)</li> <li>• ProtocolSignal (O)</li> <li>• OriginRealm (C)</li> <li>• TargetNetwork (O)</li> <li>• ResultCode (C)</li> <li>• ReasonForTermination (C)</li> </ul> |

**Note**

The attribute with (M) represents mandatory, (O) represents optional, (C) represents conditionally available.

If 3GPP-IMSI is present in the incoming request packet, the following 3GPP-related attributes are also reported to the LI server:

- 3GPP-IMSI
- Called-Station-Id
- Calling-Station-Id
- 3GPP-PDP-Type
- SGSN-Address

- GGSN-Address
- 3GPP-IMSI-MCC-MNC
- 3GPP-NSAPI
- 3GPP-SGSN-MCC-MNC
- 3GPP-IMEISV

### Stopping Monitoring Process

On receiving the DeprovisionTarget request from LIS, the target user is removed from the monitoring list.

### Querying Target User Events

On receiving the LinkUpdate request on target user from LIS, the target user details are checked in the monitoring list and message is sent to LIS as listed below:

- If the specified user is not currently being monitored, a reply with reason-code indicating that the user is currently not targeted is sent.
- If the specified user is currently being targeted and is not logged into the network, a reply with status stating that the user is “inactive” in the network is sent.
- If the specified user is currently being targeted and is logged into the network, a reply with the following attributes is sent:
  - Case Identity (M)
  - IAP System Identity (M)
  - Time Stamp (M)
  - Subscriber Identity (M)
  - Access Method (C)
  - Network Access Node Identity (C)
  - IP address (C)
  - Access Session Identity (M)
  - Access Session Characteristics (C)
  - Location information (C)
  - Protocol Signal (O)

### Viewing Monitored Users

On receiving the ListTarget request from LIS, a list of users that are currently being monitored are sent to LIS. The reply will contain a surveillance-target-count attribute indicating the count of the number of users being targeted and multiple instances of surveillance-target-identifier attribute having the real identifiers.

### Intercept Response

Each request from the LIS contains a transaction-id which is copied on to the reply from Prime Access Registrar. For each request type there is an appropriate response type with appropriate return data.

Table 9-9 provides the list of Intercept results for RADIUS and Diameter.

**Table 9-11 Intercept Results for RADIUS and Diameter**

| Intercept Response (RADIUS) | Intercept Response (Diameter) | Description  |
|-----------------------------|-------------------------------|--|
| ProvisionTargetResult       | DiaProvisionTargetResult      | An acknowledgment for the request with the same transaction ID. For information on the request, see <a href="#">Initiating Monitoring Process, page 9-52</a> . |
| DeprovisionTargetResult     | DiaDeProvisionTargetResult    | An acknowledgment for the request with the same transaction ID. For information of the request, see <a href="#">Stopping Monitoring Process, page 9-55</a> .   |
| LinkUpdateResult            | DiaLinkUpdateResult           | For LinkUpdate, see <a href="#">Querying Target User Events, page 9-55</a> .   |
| ListTargetResult            | DiaListTargetResult           | For ListTarget, see <a href="#">Viewing Monitored Users, page 9-55</a> .   |

## Configuring Lawful Intercept

Two scripts which are LawfulIntercept and RexLiScript are to be configured to run LawfulIntercept service in Prime Access Registrar. LawfulIntercept script should be configured in the server's incoming scripting point which is used to check the provisioned status of the user in the incoming access request. If the user is provisioned in the data store, Virtual-Server-Outgoing-Script will be executed after the server's incoming point.

InitEntryPoint of LawfulIntercept script writes the targeted list of users to a file while the server is stopping and reads the targeted users back to data store while the server is starting.

RexLiScript is configured in Virtual-Server-Outgoing-Script that sends events of the provisioned users to the LI service client.

### Configuring the Lawful Intercept

To configure Lawful Intercept:

- Step 1** Create the RexLiScript script object that will be set in Virtual-Server-Outgoing-Script point.

```
[ //localhost/Radius/Scripts/virtual ]
  Name = virtual
  Description =
  Language = rex
  Filename = libLiScript.so
  EntryPoint = RexLiScript
  InitEntryPoint = InitRexLiScript
  InitEntryPointArgs =
```

- Step 2** Create the LawfulIntercept script object.

```
[ //localhost/Radius/Scripts/LiScript ]
  Name = LiScript
  Description =
  Language = Rex
  Filename = libLiScript.so
  EntryPoint = LawfulIntercept
  InitEntryPoint = RexInitialize
  InitEntryPointArgs = virtual
```

**Step 3** set LawfulIntercept script object to ServerIncoming scripting point;

```
[ //localhost/Radius ]
  IncomingScript~ = LiScript
```



**Note** The file 'libLiScript.so' comes up with Prime Access Registrar kit. You have to copy it into /cisco-ar/scripts/radius/rex/ path.

**Step 4** Save the configuration:

```
save
```

**Step 5** Reload the configuration:

```
reload
```

## TACACS+ Support for AAA

TACACS+ (Terminal Access Controller Access-Control System Plus) is a terminal access control protocol for routers, switches, network access servers and other networked computing devices. The main goal of TACACS+ is to provide separate authentication, authorization and accounting services.

In Prime Access Registrar, TACACS+ supports authentication, command authorization, and accounting. The authentication support is available for login services with PAP, CHAP, and ASCII authentication types. It also tracks and maintains the executed command details in the command accounting database. Configuration is supported through the CLI/GUI and statistics are provided through CLI, GUI, and SNMP. TACACS+ supports the following Prime Access Registrar services:

- Local-users and Local-file service
- OCI
- ODBC
- LDAP

The following shows an example configuration for TACACS+:

```
[ /Radius/Clients/mytac ]
  Name = mytac
  Description =
  Protocol = tacacs-and-radius
  IPAddress = 10.77.123.57
  SharedSecret = <encrypted>
  Type = NAS
  Vendor =
  IncomingScript~ =
  OutgoingScript~ =
  EnableDynamicAuthorization = FALSE
  NetMask =
  EnableNotifications = FALSE
  EnforceTrafficThrottling = TRUE
```

Prime Access Registrar provides command authorization support to authorize the cmd mode commands. Command authorization is based on device access rules and the decision to authorize is based on command sets and conditions or expressions defined for the access rules. They determine whether to authorize a set of commands for the user or not.

If you enable TACACS+ command authorization for a service, you must define the following:

- Command sets—You must configure the list of commands with the arguments and the action to perform: permit or deny.
- Device access rules—You must configure the conditions or expressions and the command sets that are applicable to the access rule if the conditions are met.
- Service—You must enable the device access and associate the device access rules for the service.

When a packet enters the service, it selects the first device access rule and evaluates the condition. If the condition is met, then the service applies the device access rule for the request. If the command that is processed matches a command listed in the command set, the service decides on whether to permit the command for the user or not based on the permissions set up. See the example below.

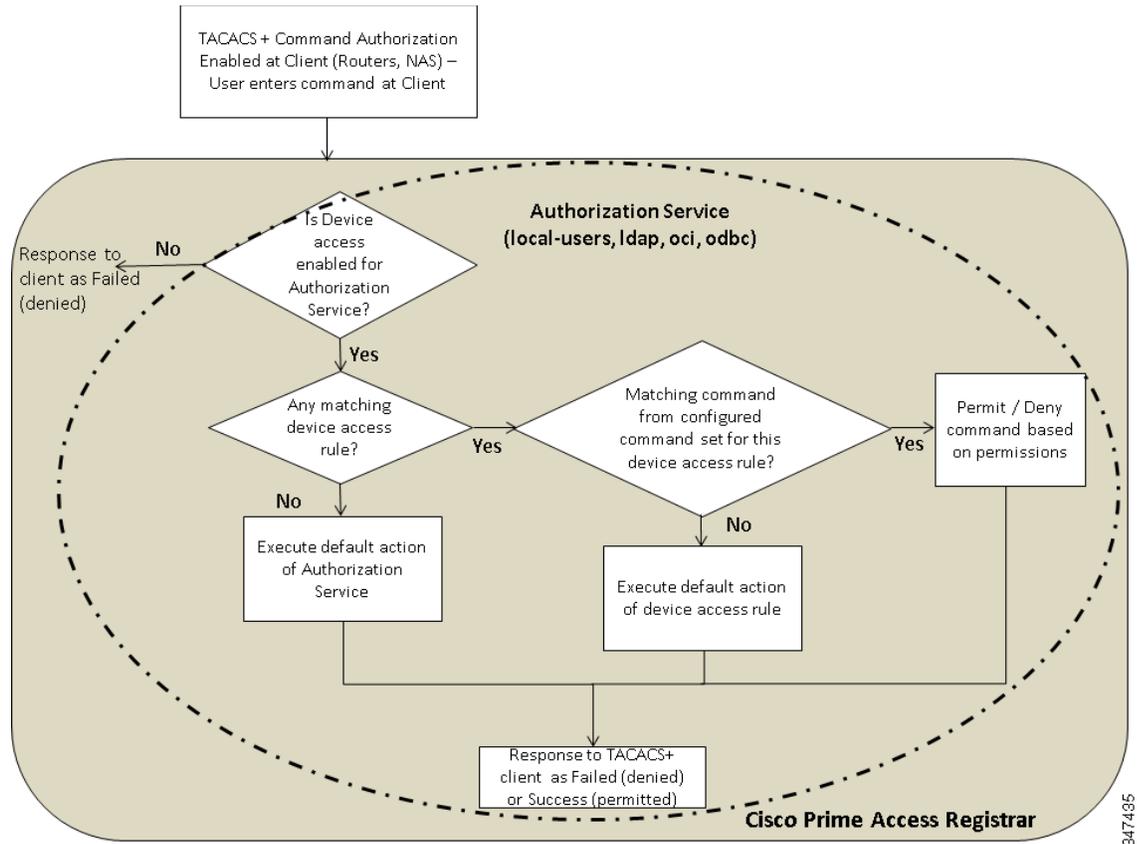
| Device Access Rule | Condition   | Command Set | Command | Arguments  | Action |
|--------------------|---|-------------|---------|------------|--------|
| NewAccessRule      | Expr1 OR Expr2<br>Where:<br>Expr1 = user-name=bob<br>Expr2 = nas-identifier=~PGW*/<br>OR = Conditional operator | cmdset1     | show    | *          | permit |
|                    |   |             | enable  | ~/serial*/ | deny   |

In the above example, if one of the conditions user-name = bob or nas-identifier = ~/PGW\*/ is met, then the service applies the device access rule. If the processed command with its arguments matches one of the commands listed above, then the service permits or denies the command according to the setup.

**Note** Prime Access Registrar supports POSIX Extended Regular Expression (ERE) for command arguments and condition expressions value property.

Figure 9-2 shows the transaction flow for TACACS+ command authorization.

Figure 9-2 TACACS+ Command Authorization Flow



The following is an example configuration of device access rules and command sets configured for a local-users service:

```

[ //localhost/Radius ]
  Name = Radius
  Description =
  Version = 7.2.0.0
  IncomingScript~ =
  OutgoingScript~ =
  DefaultAuthenticationService~ = local-users
  DefaultAuthorizationService~ = local-users
  DefaultAccountingService~ = local-file
  DefaultSessionService~ =
  DefaultSessionManager~ = session-mgr-1
  UserLists/
  UserGroups/
  Policies/
  Clients/
  Vendors/
  Scripts/
  Services/
  SessionManagers/
  ResourceManagers/
  Profiles/
  Rules/
  Translations/
  TranslationGroups/
  RemoteServers/
  CommandSets/

```

```

DeviceAccessRules/
FastRules/
Advanced/
Replication/

--> cd /r/DeviceAccessRules/

[ //localhost/Radius/DeviceAccessRules ]
  Entries 0 to 0 from 0 total entries
  Current filter: <all>

--> add d2

Added d2

--> cd d2

[ //localhost/Radius/DeviceAccessRules/d2 ]
  Name = d2
  Description =
  CommandSetNames =
  Conditions =
  DefaultDeviceAction = PermitAll
  ConditionExpressions/

--> set Conditions "A1 and A2"

Set Conditions "A1 and A2"

--> SET CommandSetNames "cm1, CM2"

Set CommandSetNames "cm1, CM2"

--> CD ConditionExpressions/

[ //localhost/Radius/DeviceAccessRules/d2/ConditionExpressions ]
  Entries 0 to 0 from 0 total entries
  Current filter: <all>

--> add a1

Added a1

--> add a2

Added a2

--> cd a1

[ //localhost/Radius/DeviceAccessRules/d2/ConditionExpressions/a1 ]
  Name = a1
  Description =
  Attribute =
  Value =

--> Set Attribute user-name

Set Attribute user-name

--> Set Value user*

Set Value user*

```

```
--> cd ..

[ //localhost/Radius/DeviceAccessRules/d2/ConditionExpressions ]
  Entries 1 to 2 from 2 total entries
  Current filter: <all>

  a1/
  a2/

--> cd a2

[ //localhost/Radius/DeviceAccessRules/d2/ConditionExpressions/a2 ]
  Name = a2
  Description =
  Attribute =
  Value =

--> Set Attribute user-group

Set Attribute user-group

--> Set Value ABC

Set Value ABC

--> cd /r/CommandSets/

[ //localhost/Radius/CommandSets ]
  Entries 0 to 0 from 0 total entries
  Current filter: <all>

--> add cm1

Added cm1

--> cd cm1

[ //localhost/Radius/CommandSets/cm1 ]
  Name = cm1
  Description =
  Commands/

--> cd Commands/

[ //localhost/Radius/CommandSets/cm1/Commands ]

--> Set 1 "permit show *"

Set 1 "permit show *"

--> cd ..

[ //localhost/Radius/CommandSets/cm1 ]
  Name = cm1
  Description =
  Commands/

--> cd ..

[ //localhost/Radius/CommandSets ]
  Entries 1 to 1 from 1 total entries
  Current filter: <all>
```

```
cm1/
--> add cm2

Added cm2

--> cd cm2

[ //localhost/Radius/CommandSets/cm2 ]
  Name = cm2
  Description =
  Commands/

--> cd commands/

[ //localhost/Radius/CommandSets/cm2/Commands ]

--> Set 1 "deny show all"

Set 1 "deny show all"
--> sav

Validating //localhost...
Saving //localhost...
```

For more information on configuring the command sets and device access rules in the GUI, see the [CommandSets, page 2-57](#) and [DeviceAccessRules, page 2-58](#) sections in [Chapter 2, “Using the Graphical User Interface.”](#)



## Directing RADIUS Requests

---

You can use the policy engine to determine the AAA services for processing a request packet based on the User-Name suffix, User-Name prefix, Calling-Station-ID, Called-Station-ID and Nas-IP-Address. You configure the policy Engine through policies and rules.

This chapter contains the following sections:

- [Configuring Policies and Rules](#)
- [Routing Requests](#)
- [Standard Scripts Used with Rules](#)

## Configuring Policies and Rules

A policy is a group of rules. Each rule consists of a set of conditions and corresponding services. A rule succeeds if all the conditions specified in the rule are satisfied. If a rule succeeds, the services indicated by its service attributes are used to process the packet. However, Prime Access Registrar defers packet processing until the policy succeeds.

This section contains the following topics:

- [Configuring Policies](#)
- [Configuring Rules](#)
- [Wildcard Support](#)
- [Script and Attribute Requirements](#)
- [Validation](#)
- [Known Anomalies](#)

## Configuring Policies

You configure policies under **/Radius/Policies**. To enable the Prime Access Registrar server to use policies, you must first configure policy named SelectPolicy.

```
[ //localhost/Radius/Policies/SelectPolicy ]
  Name = SelectPolicy
  Description =
  Grouping = rule1|rule2
```

The Grouping property of a policy determines which rules are to be evaluated and in which order. Rules are evaluated from left to right. Use the pipe (|) or ampersand (&) character to group rules.

**Note**

Before you can provide rules in the Grouping property, the rules must first be added to the configuration under **/Radius/Rules**.

The following are the Grouping property rules:

- If rules are grouped with the pipe character (`rule1|rule2`), all rules in the group are evaluated in sequential order until one of the rules succeeds. If any one of the rules in the policy succeeds, the policy succeeds.
- If rules are grouped with the ampersand character (`rule1&rule2&rule3`), all the rules listed are evaluated until one of the rules fails. For the policy to succeed, all the rules in the policy must succeed.

## Configuring Rules

You configure rules under **/Radius/Rules**. When you add a rule, provide the script that should be executed for the rule and the attributes to use if the rule succeeds. The script you specify must be defined under **/Radius/Scripts**, as shown in the following:

```
[ //localhost/Radius/Rules/rule1 ]
  Name = rule1
  Description =
  Type = radius
  Script~ =
  Attributes/
    Authentication-service = local-users
    Authorization-service = local-users
    Realm = @cisco.com

[ //localhost/Radius/Scripts/ExecRealmRule ]
  Name = ExecRealmRule
  Description =
  Language = Rex
  Filename = librexscript.so
  EntryPoint = ExecRealmRule
  InitEntryPoint =
  InitEntryPointArgs =
```

## Wildcard Support

Prime Access Registrar supports limited wildcard functionality in rules for Realm, DNIS, and CLID attributes, specifically the asterisk (\*) and question mark (?) characters. The asterisk matches any number of characters, including the null character. The question mark matches any single character, not including the null character. Prime Access Registrar also supports both wildcard characters in one pattern, as in `CLID = 180098?87*`.

**Note**

The realms should start with either the @ or # character. For example, `Realm=@cisco.com`.

- For an exact matching of the realm, you should configure the rule with the exact realm. For example, for an exact match to `abc@cisco.com`, you should use `Realm=@cisco.com`.

- If you use Realm=cisco.com (without any valid character), values such as xyz@us.cisco.com, xyz@uk.cisco.com, abc#cisco.com, and so on can also match and return a success.

The following is an example using the asterisk wildcard character used in a Rule named *rule1*:

```
[ //localhost/Radius/Rules/rule1 ]
  Name=rule1
  Description =
  ScriptName = ExecRealmRule
  Attributes/
    Authentication-Service = Local-Users
    Authorization-Service = Local-Users
    Realm = ~/@*cisco.com/
```

Rule *rule1* succeeds when the domain of the username in an access request matches the *@\*cisco.com* pattern. Each of the following is a good match: *@us.cisco.com*, *@eng.cisco.com*, and *@cisco.com*. With a match, the **ExecRealmRule** script sets Authentication-Service and Authorization-Service to Local-Users in the environment dictionary.

The following is an example using the "?" wildcard character in a Rule named *rule2*:

```
[ //localhost/Radius/Rules/rule2 ]
  Name = rule2
  Description =
  ScriptName = ExecDNISRule
  Attributes/
    Authentication-Service = Translation-Service
    Authorization-Service = Translation-Service
    DNIS = 1800345987?
```

Rule *rule2* succeeds if the Called-Station-Id attribute (DNIS) in the packet matches 1800345987?. Each of the following is a good match: 18003459876 and 18003459870, while 1800345987 is not. With a match, the **ExecDNISRule** script sets Authentication-Service and Authorization-Service to Translation-Service in the environment dictionary.

## Script and Attribute Requirements

The following script and attribute requirements exist:

- **/Radius/Policies/SelectPolicy** is the first policy Prime Access Registrar applies.
- The characters "|" and "&" are reserved as logical operands in a Grouping definition; they cannot be included in a **/Radius/Rules** name.
- A space is not permitted between the logical operands and the rules in a Grouping definition.
- The scripts included in the rules must be defined under the **/Radius/Scripts** directory.
- The attributes included in the rules must be defined under the **/Radius/Advanced/Attribute Dictionary** directory.
- The rules included in the policies must be defined under the **/Radius/Rules** directory.

## Validation

When policies are configured, Prime Access Registrar performs the following validations:

- Ensures the scripts included in the rules are defined under the **/Radius/Scripts** directory.
- Ensures the attributes included in the rules are defined under the **/Radius/Advanced/Attribute Dictionary** directory.
- Ensures the rules included in the policies are defined under the **/Radius/Rule** directory.

## Known Anomalies

The following anomalies currently exist:

- Grouping expressions are not checked for validity.
- The use of parentheses to denote precedence is not supported in a Grouping definition.
- A check is not performed to determine whether a policy that is included within another policy is defined under the **/Radius/Policies** directory.

## Routing Requests

Using the policy engine, Prime Access Registrar enables you to route requests based on attributes in access request packets. The following sections describe how to route requests based on different attributes:

- [Routing Requests Based on Realm](#)
- [Routing Requests Based on DNIS](#)
- [Routing Requests Based on CLID](#)
- [Routing Requests Based on NASIP](#)
- [Routing Requests Based on User-Name Prefix](#)
- [Attribute Translation](#)
- [Time of Day Access Restrictions](#)

### Routing Requests Based on Realm

The Prime Access Registrar policy engine can process request packets based on the realm in the User-Name attribute.

In the following example, request packets with the User-Name attribute containing *@abc.com* as the suffix should be processed locally and the request packets with User-Name attribute containing *@xyz.com* should be proxied to a remote AAA Server.

```
[ //localhost/Radius/Policies ]
  SelectPolicy/
    Name = SelectPolicy
    Description =
    Grouping = abcrule|xyzrule
```

The following SelectPolicy refers to two rules *abcrule* and *xyzrule*:

1. When a request packet arrives, Prime Access Registrar executes SelectPolicy beginning with *abcrule* to determine if the User-Name attribute contains *@abc.com* as the realm. If so, the *abcrule* is successful as is SelectPolicy, therefore the packet is processed locally.
2. If the User-Name attribute does not contain *@abc.com* as the realm, Prime Access Registrar executes *xyzrule* to determine if the User-Name attribute contains *@xyz.com*. If so, *xyzrule* is successful as is SelectPolicy. Hence the request is proxied to the remote server specified in *xyz-service*.

In this example, the rules are grouped using the | (or) operator. So all the rules specified in the grouping parameter will be executed until one of them succeeds.

```
[ //localhost/Radius/Rules ]
  abcrule/
    Name = abcrule
    Description =
    Script~ = ExecRealmRule
    Attributes/
      Authentication-Service = local-users
      Authorization-Service = local-users
      Realm = @abc.com

  xyzrule/
    Name = xyzrule
    Description =
    Script~ = ExecRealmRule
    Attributes/
      Authentication-Service = xyz-service
      Authorization-Service = xyz-service
      Realm = @xyz.com
```

The ExecRealmRule script matches the realm with the suffix in the User-Name attribute and sets the appropriate service for processing the packet. This is a standard script available with Prime Access Registrar. Prime Access Registrar can also be configured to set a particular kind of service for multiple realms. For example, the following configuration can be used if packets with *@pqr.com* or *@klm.com* should be processed using the same service *klm-service*.

```
[ //localhost/Radius/Rules ]
  rulex/
    Name = rulex
    Description =
    Script~ = ExecRealmRule
    Attributes/
      Authentication-Service = klm-service
      Authorization-Service = klm-service
      Realm = "@pqr.com" "@klm.com"
```

## Routing Requests Based on DNIS

The Prime Access Registrar policy engine can process request packets differently based on the DNIS (Called-Station-Id) attribute in the request packet.

In the following example, request packets with the Calling-Station-Id attribute that contain 1111111 should be processed by *abc-service*, while request packets with the Called-Station-Id attribute that contain 2222222 or 3333333 should be processed using *xyz-service*.

```
[ //localhost/Radius/Policies ]
  SelectPolicy/
```

```
Name = SelectPolicy
Description =
Grouping = abcrule|xyzrule
```

The following SelectPolicy refers to two rules, *abcrule* and *xyzrule*:

1. When a request packet arrives, Prime Access Registrar executes SelectPolicy beginning with abcrule to determine if the DNIS attribute contains 1111111. If so, the abcrule is successful as is SelectPolicy, and the packet is processed using abc-service.
2. If the Called-Station-Id attribute does not contain 1111111, Prime Access Registrar executes the xyzrule to determine if the Called-Station-Id attribute contains 2222222 or 3333333. If so, xyzrule is successful as is SelectPolicy, and the packet is processed using xyz-service.

```
[ //localhost/Radius/Rules ]
abcrule/
  Name = abcrule
  Description =
  Script~ = ExecDNISRule
  Attributes/
    Authentication-Service = abc-service
    Authorization-Service = abc-service
    DNIS = 1111111

xyzrule/
  Name = xyzrule
  Description =
  Script~ = ExecDNISRule
  Attributes/
    Authentication-Service = xyz-service
    Authorization-Service = xyz-service
    DNIS = "2222222" "3333333"
```

The **ExecDNISRule** script matches the DNIS value configured in Prime Access Registrar with the value in the Called-Station-Id attribute of the request packet and sets the appropriate service for processing the packet. **ExecDNISRule** is a standard script available with Prime Access Registrar.

## Routing Requests Based on CLID

The Prime Access Registrar policy engine can process request packets differently based on the CLID value in arriving request packets.

In the following example, the request packets with a Calling-Station-Id (CLID) attribute value of 1111111 should be processed by abc-service and the request packets with the CLID attribute value of 2222222 or 3333333 should be processed using xyz-service.

```
[ //localhost/Radius/Policies ]
SelectPolicy/
  Name = SelectPolicy
  Description =
  Grouping = abcrule|xyzrule
```

The following SelectPolicy refers to two rules, *abcrule* and *xyzrule*:

1. When a request packet arrives, Prime Access Registrar executes SelectPolicy beginning with abcrule to determine if the CLID attribute contains 1111111. If so, the abcrule is successful as is SelectPolicy, and the packet is processed using abc-service.
2. If the CLID attribute does not contain 1111111, Prime Access Registrar executes xyzrule to determine if the CLID attribute contains 2222222 or 3333333. If so, xyzrule is successful and hence SelectPolicy becomes successful and the packet is processed using xyz-service.

```
[ //localhost/Radius/Rules ]
  abcrule/
    Name = abcrule
    Description =
    Script~ = ExecCLIDRule
    Attributes/
      Authentication-Service = abc-service
      Authorization-Service = abc-service
      CLID = 1111111

  xyzrule/
    Name = xyzrule
    Description =
    Script~ = ExecCLIDRule
    Attributes/
      Authentication-Service = xyz-service
      Authorization-Service = xyz-service
      CLID = "2222222" "3333333"
```

The **ExecCLIDRule** script matches the CLID value configured in Prime Access Registrar with the value in the CLID attribute of the request packet and sets the appropriate service for processing the packet. **ExecCLIDRule** is a standard script available with Prime Access Registrar.

## Routing Requests Based on NASIP

The Prime Access Registrar policy engine can process request packets differently based on the client IP address value in arriving request packets.

In the following example, arriving request packets with the NAS-IP-Address attribute value 1.1.1.1 should be processed by abc-service and arriving request packets with the NAS-IP-Address attribute value 2.2.2.2 should be processed using xyz-service.

```
[ //localhost/Radius/Policies ]
  SelectPolicy/
    Name = SelectPolicy
    Description =
    Grouping = abcrule|xyzrule
```

The following SelectPolicy refers to two rules, *abcrule* and *xyzrule*:

1. When a request packet arrives, Prime Access Registrar executes SelectPolicy beginning with abcrule to determine if the NAS-IP-Address attribute contains an IP address from the subnet 1.1.1.0/24. If so, the abcrule is successful as is SelectPolicy, and the packet is processed using abc-service.
2. If the NAS-IP-Address attribute does not contain an IP address from the subnet 1.1.1.0/24, Prime Access Registrar executes xyzrule to determine if the NAS-IP-Address attribute contains 2.2.2.2. If so, xyzrule is successful as is SelectPolicy, and the packet is processed using xyz-service.

```
[ //localhost/Radius/Rules ]s
  abcrule/
    Name = abcrule
    Description =
    Script~ = ExecNASIPRule
    Attributes/
      Authentication-Service = abc-service
      Authorization-Service = abc-service
      Client-IP-Address = 1.1.1.0
      Subnet-mask = 255.255.255.0

  xyzrule/
```

```
Name = xyzrule
Description =
Script~ = ExecNASIPRule
Attributes/
  Authentication-Service = xyz-service
  Authorization-Service = xyz-service
  Client-IP-Address = 2.2.2.2
```

The **ExecNASIPRule** script matches the Client IP address configured in Prime Access Registrar with the value in the NAS-IP-Address attribute of the request packet and sets the appropriate service for processing the packet. **ExecNASIPRule** is a standard script available with Prime Access Registrar.

## Routing Requests Based on User-Name Prefix

You can use the Prime Access Registrar policy engine to select a service based on the prefix in the User-Name attribute.

In the following example, request packets with a User-Name attribute that contains @abc.com as the suffix and cisco as the prefix should be processed using the service abc-service. A request packet with User-Name attribute containing cisco/bob@abc.com will be processed using abc-service.

```
[ //localhost/Radius/Policies ]
  SelectPolicy/
    Name = SelectPolicy
    Description =
    Grouping = suffixrule & prefixrule
```

The following SelectPolicy refers to two rules, *suffixrule* and *prefixrule*:

1. When a request packet arrives, Prime Access Registrar executes SelectPolicy beginning with *suffixrule* to determine if the realm in the User-Name attribute contains @abc.com. If so, the *suffixrule* is successful. Since there is an “&” operator between the rules, the *prefixrule* must also succeed for the SelectPolicy to be successful.
2. The *prefixrule* is now processed to determine if the prefix in the User-Name attribute contains cisco. If so, the *prefixrule* is successful which makes SelectPolicy successful, and the AA service is set to the service specified in the *prefixrule*.

```
[ //localhost/Radius/Rules ]
  abcrule/
    Name = suffixrule
    Description =
    Script~ = ExecRealmRule
    Attributes/
      Realm = @abc.com

  prefixrule/
    Name = prefixrule
    Description =
    Script~ = ExecPrefixRule
    Attributes/
      Authentication-Service = abc-service
      Authorization-Service = abc-service
      Delimiters = @#%&/
      Prefix = cisco
      StripPrefix = No
```

**ExecPrefixRule** script matches the prefix configured in Prime Access Registrar with the prefix in the User-Name attribute of the request packet and sets the appropriate service for processing the packet. **ExecPrefixRule** is a standard script available with Prime Access Registrar. See [ExecPrefixRule](#) for more information.

## Attribute Translation

The attribute translation feature supports the RADIUS proxy enabling you to customize attribute filters so that RADIUS attribute value (AV) pairs can be inserted, deleted, or substituted.

For example, when a request is proxied from AAA Server on ISP A to AAA Server on ISP B, some AV pairs might be substituted (such as IP address) because they might not be valid on the ISP B network. Additionally, ISP B might return some vendor-specific attributes (VSAs) that are not applicable to ISP A's network. Therefore, ISP A will substitute ISP B's VSA value pairs for ISP A's VSAs.

Two configuration points under the **/Radius** directory support this feature,

- [Translations](#)
- [TranslationGroups](#)
- [Parsing Translation Groups](#)

## Translations

Under the **/Radius/Translations** directory, any translation to insert, substitute, or translate attributes can be added. The following is a sample configuration under the **/Radius/Translations** directory:

```
[ //localhost/Radius/Translations/T1 ]
  Name = T1
  Description =
  DeleteAttrs = Session-Timeout, Called-station-id
  Attributes/
    Calling-Station-id = 1232909
```

DeleteAttrs is the set of attributes to be deleted from the packet. Each attribute is comma separated and no spaces are allowed between attributes.

Under the **/Radius/Translations/T1/Attributes** directory, the attributes that should be inserted and the attributes that should be substituted are specified. These AV pairs are either added to the packet if not present already or replaced with the configured value.

## TranslationGroups

Under the **/Radius/TranslationGroups** directory, translations can be grouped and applied to certain sets of packets, which are referred to in a rule.

The following is a sample configuration under the **/Radius/TranslationGroups** directory:

```
[ //localhost/Radius/TranslationGroups/CiscoIncoming ]
  Name = CiscoIncoming
  Description =
  Translations/
    1. T1
```

The translation group is referenced through the Prime Access Registrar policy engine in the **/Radius/Rules/<RuleName>/Attributes** directory.

- Incoming-Translation-Groups are set to a translation group (for example CiscoIncoming).

- `Outgoing-Translation-Groups` are to set to another translation group (for example `CiscoOutgoing`).

The following is an example of setting up a rule for a translation group.

```
[ //localhost/Radius/Rules/ciscotranslationrule ]
  Name = ciscotranslationrule
  Description =
  Script~ = ParseTranslationGroupsByRealm
  Attributes/
    Incoming-Translation-Groups = CiscoIncoming
    Outgoing-Translation-Groups = CiscoOutgoing
    Realm = @cisco.com
```

The `ciscoTranslationRule` rule must be referred to in the `/Radius/Policies` directory, so the Prime Access Registrar policy engine can invoke this rule. If the pattern of `Realm`, `DNIS`, or `CLID` matches the one defined in the rule, Prime Access Registrar sets the environment variable `Incoming-Translation-Groups` to `CiscoIncoming`. By looking up the definition of `CiscoIncoming`, Prime Access Registrar applies all of the translations to the incoming packet (before it is proxied to the other server).

When the proxied packet comes back to the RADIUS server, Prime Access Registrar sets the environment variable, `Outgoing-Translation-Groups` to `CiscoOutgoing`.

`DNIS`, `CLID`, and `Realm` are supported for filtering packets. Prime Access Registrar provides the following scripts to facilitate filtering based on `DNIS`, `CLID` and `Realm`.

## Parsing Translation Groups

Prime Access Registrar provides three scripts that enable you to parse translation groups based on the `DNIS`, `CLID` or `Realm` attribute in an incoming packet. These scripts are:

- `ParseTranslationGroupsByDNIS`
- `ParseTranslationGroupsByCLID`
- `ParseTranslationGroupsByRealm`

In the following example, request packets containing `@abc.com` as the realm should be proxied to the remote server defined under `abc-service`. Before redirecting the request packet to the remote server, the `Calling-Station-Id` of the packet should be changed to `111`.

```
[ //localhost/Radius/Policies ]
  SelectPolicy/
    Name = SelectPolicy
    Description =
    Grouping = realmrule & translaterule
```

The following `SelectPolicy` refers to two rules, `realmrule` and `translaterule`:

1. When a request packet arrives, Prime Access Registrar executes `SelectPolicy` beginning with “`realmrule`” to determine if the realm in the `User-Name` attribute contains `1.1.1.1`. If so, the `realmrule` is successful and the AA service is set to `abc-service`.
2. Next Prime Access Registrar executes the `translaterule` to change the `CLID` of the packet to `111`.

```
[ //localhost/Radius/Rules/ciscotranslationrule ]
  Name = ciscotranslationrule
  Description =
  Script~ = ParseTranslationGroupsByRealm
  Attributes/
    Incoming-Translation-Groups = CiscoIncoming
    Realm = @cisco.com
```

```
[ //localhost/Radius/Translations ]
  Entries 1 to 1 from 1 total entries
  Current filter: <all>
  T1/
    Name = T1
    Description =
    Attributes/
      calling-station-id = 111

[ //localhost/Radius/TranslationGroups ]
  Entries 1 to 1 from 1 total entries
  Current filter: <all>
  CiscoIncoming/
    Name = CiscoIncoming
    Description =
  Translations/
    1. T1
```

## Time of Day Access Restrictions

You can use the Prime Access Registrar policy engine to implement access restriction on users based on the time of day. The following are **ExecTimeRule** script that implements this functionality:

- **ExecTimeRule** can be used to check the time at which the request packet arrives and determine if access should be granted based on the authorization parameters for the user.
- If the rule succeeds, **ExecTimeRule** sets the Acceptedprofiles Environment dictionary variable to a profile or a set of profiles, as in the following:

```
Acceptedprofiles=Regularaccess::Highprivilegeaccess
```



### Note

If more than one profile is to be added to the Acceptedprofiles variable, use two colons to separate them (::).

If the user is authenticated, the Baseprofile of the user is compared with the Acceptedprofiles. All the profiles that are in the Baseprofile and in Acceptedprofiles will be used as profiles while sending the response for the user.

For example, consider the following user configuration of user1:

```
[ //localhost/Radius/UserLists/new/user1 ]
  Name = user1
  Description =
  Password = <encrypted>
  AllowNullPassword = FALSE
  Enabled = TRUE
  Group~ = regularusers
  BaseProfile~ =highprivilegeaccess::readonlyaccess::regularaccess
  AuthenticationScript~ =
  AuthorizationScript~ =
  UserDefined1 =
  Attributes/
  CheckItems/
```

The Baseprofile of the user1 has highprivilegeaccess, readonlyaccess and regularaccess. If the Acceptedprofiles of the user has regularaccess and highprivilegeaccess, the profiles regularaccess and highprivilegeaccess will be used while sending the response packet.

This section contains the following topics:

- [Setting Time Ranges in ExecTimeRule](#)
- [ExecTimeRule Example Configuration](#)
- [Reducing Overhead Using Policies to Group Rules](#)
- [ParseTranslationGroupsByDNIS](#)

## Setting Time Ranges in ExecTimeRule

ExecTimeRule accepts time range in the following format.

**Set timerange “\* \* \* \* \*”**

The first star indicates minutes and can be a value from 0-59. The second star indicates hours and can be a value from 0-23. The third star indicates day of the month and can be a value from 1-31. The fourth star indicates month and can be a value from 1-12. The fifth star indicates day of the week and can be a value from 0-6 where 0 indicates Sunday, 1 indicates Monday, and so on.

For example, to schedule a particular action to occur every Sunday during the month of December, use a command line like this:

**Set timerange “\* \* \* 12 0”**

## ExecTimeRule Example Configuration

This section provides a configuration example where a user, user1, is only authorized for PPP service between 10 AM and 6 PM. If a login occurs at any other time, user1 will be authorized only for telnet service.

### Policies

```
[ //localhost/Radius/Policies ]
Entries 1 to 1 from 1 total entries
Current filter: <all>
SelectPolicy/
  Name = SelectPolicy
  Description =
  Grouping = ppprule|telnetrule
```

### Rules

```
[ //localhost/Radius/Rules ]
Entries 1 to 2 from 2 total entries
Current filter: <all>
ppprule/
  Name = ppprule
  Description =
  Script~ = ExecTimeRule
  Attributes/
    acceptedprofiles = default-ppp-users
    timerange = "* 10-18 * * * "
telnetrule/
  Name = telnetrule
  Description =
  Script~ = ExecTimeRule
  Attributes/
    acceptedprofiles = default-telnet-users
```

```
timerange = "* 0-10,18-23 * * * "
```

## Profiles

```
[ //localhost/Radius/Profiles ]
Entries 1 to 5 from 5 total entries
Current filter: <all>
default-PPP-users/
  Name = default-PPP-users
  Description =
  Attributes/
    Ascend-Idle-Limit = 1800
    Framed-Compression = "VJ TCP/IP header compression"
    Framed-MTU = 1500
    Framed-Protocol = PPP
    Framed-Routing = None
    Service-Type = Framed
default-Telnet-users/
  Name = default-Telnet-users
  Description =
  Attributes/
    Login-IP-Host = 204.253.96.3
    Login-Service = Telnet
    Login-TCP-Port = 541
```

## User

```
[ //localhost/Radius/UserLists/new/user1 ]
Name = user1
Description =
Password = <encrypted>
AllowNullPassword = FALSE
Enabled = TRUE
Group~ = regularusers
BaseProfile~ = default-telnet-users::default-ppp-users
AuthenticationScript~ =
AuthorizationScript~ =
UserDefined1 =
Attributes/
CheckItems/
```

## Reducing Overhead Using Policies to Group Rules

When you configure a large number of rules, the processing of request packets can be slow. For example, if you have 50 rules and only the last rule succeeds, the Prime Access Registrar server will have to check the preceding 49 rules before executing the rule that succeeds. You can reduce this overhead by using policies to group rules.

The following sample configuration, Prime Access Registrar must choose the AA service to be used for two domains, abc.com and xyz.com, based on the DNIS. You can do this by configuring different policies for different domains.

## Policies

In the following configuration, SelectPolicy selects the policy to process packets with realm abc.com or xyz.com. Based on the realm that arrives in the request packet, abcrealmrule and xyzrealmrule decide whether to use abc-policy or xyz-policy to process the packets. abc-policy and xyz-policy are configured with rules to check for DNIS numbers in the respective domains and set the AA services appropriately.

```
[ //localhost/Radius/Policies ]
Entries 1 to 3 from 3 total entries
Current filter: <all>
SelectPolicy/
  Name = selectpolicy
  Description =
  Grouping = abcrealmrule|xyzrealmrule
abc-policy/
  Name = abc-policy
  Description =
  Grouping = abcDNISrule1|abcDNISrule2
xyz-policy/
  Name = xyz-policy
  Description =
  Grouping = xyzDNISrule1|xyzDNISrule2
```

## Rules

```
[ //localhost/Radius/Rules ]
Entries 1 to 6 from 6 total entries
Current filter: <all>

abcrealmrule/
  Name = abcrealmrule
  Description =
  Script~ = ExecRealmRule
  Attributes/
    policy = abc-policy
    realm = @abc.com
xyzrealmrule/
  Name = xyzrealmrule
  Description =
  Script~ = ExecRealmRule
  Attributes/
    policy = xyz-policy
    realm = @xyz.com
abcDNISrule1/
  Name = abcDNISrule1
  Description =
  Script~ = ExecDNISRule
  Attributes/
    Authentication-Service = abc1-service
    Authorization-Service = abc1-service
    DNIS = 1111111
abcDNISrule2/
  Name = abcDNISrule2
  Description =
  Script~ = ExecRealmRule
  Attributes/
    Authentication-Service = abc2-service
    Authorization-Service = abc2-service
    DNIS = 2222222
xyzDNISrule1/
  Name = xyzDNISrule1
  Description =
```

```

Script~ = ExecRealmRule
Attributes/
  Authentication-Service = xyz1-service
  Authorization-Service = xyz2-service
  DNIS = 6666666
xyzDNISrule2/
Name = xyzDNISrule2
Description =
Script~ = ExecRealmRule
Attributes/
  Authentication-Service = xyz2-service
  Authorization-Service = xyz2-service
  DNIS = 7777777

```

## Standard Scripts Used with Rules

Prime Access Registrar software includes the following scripts that you can use with the rules:

- [ExecRealmRule](#)
- [ExecDNISRule](#)
- [ExecCLIDRule](#)
- [ExecNASIPRule](#)
- [ExecPrefixRule](#)
- [ExecSuffixRule](#)
- [ExecTimeRule](#)
- [ParseTranslationGroupsByRealm](#)
- [ParseTranslationGroupsByDNIS](#)
- [ParseTranslationGroupsByCLID](#)

## ExecRealmRule

Use the **ExecRealmRule** script to determine the Authentication service and Authorization service to be used to process the request packet based on the suffix (Realm) in the User-Name attribute. You configure the Realm for which the packet should be checked and the service to use in the Attributes subdirectory of a rule. The **ExecRealmRule** script supports multivalued attributes with which you can configure to check for multiple Realms.

For example, the following statement checks the request packet for three realms. If one of these three realms is found in the request packet, the **ExecRealmRule** script sets the attributes to the values listed in the Attributes subdirectory of the rule that references the **ExecRealmRule** script.

```
set Realm "@cisco.com" "@foo.com" "#bar.com"
```



### Note

Only the characters @ and # can be used as delimiters in ExecRealmRule.

Prior to Cisco Prime Access Registrar (Prime Access Registrar), ExecRealmRule was interpreted as a regular expression pattern and was evaluated accordingly. ExecRealmRule now does a simple case insensitive comparison by default of the value specified for the realm attribute for the realm of a username and optionally performs regular expression matching.

You can now specify a pattern using the following notation:

*~/pattern/*

Where pattern is a string of alpha-numeric characters that might include wild card characters, as in “@\*cisco.com” to match patterns (realms) that end in *cisco.com*.



#### Note

The question mark (?) should not be used without a character pattern preceding it. Specifying ? as the first character might have undesirable results. (For regexp terminology, the question mark should be preceded by an *atom*.)

The **ExecRealmRule** script checks the request packet for the Realm and applies the values set for the following attributes:

- Authentication-Service
- Authorization-Service
- Policy

## ExecDNISRule

Use the **ExecDNISRule** script to determine the Authentication service and Authorization service to be used to process the request packet based on the Called-Station-Id (DNIS) attribute. The DNIS for which the packet should be checked and the services can be configured through the Policy Engine. The **ExecDNISRule** script supports multivalued attributes, by which you can configure multiple DNIS for checking.

For example, the following statement checks for a Calling-Station-Id of 1111111, 2222222, or 3333333. If one of the DNIS values is true, the script applies the values set for the Authentication-Service, Authorization-Service, and Policy attributes.

```
set DNIS "1111111" "2222222" "3333333"
```

## ExecCLIDRule

Use the **ExecCLIDRule** script with the Policy Engine to determine the Authentication service and Authorization service to be used to process the request packet based on the Calling-Station-Id (CLID) attribute. The CLID for which the packet should be checked and the services can be configured through the Policy Engine. **ExecCLIDRule** supports multivalued attributes by which you can configure multiple CLID for checking.

For example, the following statement checks for Calling-Station-ID and applies Authentication-Service, Authorization-Service, and Policy.

```
set CLID "1111111" "2222222" "3333333"
```

The **ExecCLIDRule** script checks the request packet for the Calling-Station-Id and applies the values set for the following attributes:

- Authentication-Service
- Authorization-Service
- Policy

## ExecNASIPRule

The Policy Engine references the **ExecNASIPRule** script to determine the AAA Services, Policy and Session Manager based on the Client-IP-Address and Subnet-Mask set in the Policy Engine. The **ExecNASIPRule** script supports multi-value attributes by which multiple you can configure the Client-IP-Address and Subnet-Mask in **aregcmd** for checking.

For example, the following statements check for Client-IP-Address and Subnet-Mask and applies Authentication-Service, Authorization-Service, Accounting-Service, Policy, and Session-Manager.

```
set Client-IP-Address "1.1.1.1" "2.2.2.2" "3.3.3.3"

set Subnet-Mask "255.255.255.0" "255.255.0.0" "255.0.0.0"
```

The **ExecNASIPRule** script checks the request packet for the Client-IP-Address and Subnet-Mask and applies the values set for the following attributes:

- Authentication-Service
- Authorization-Service
- Accounting-Service
- Policy
- Session Manager

## ExecPrefixRule

The Policy Engine references the **ExecPrefixRule** to determine the authentication and authorization services based on the prefix in the User-Name attribute of the request packet and assigns the appropriate service for processing the packet.

[Table 10-1](#) lists the **ExecPrefixRule** script attributes.

**Table 10-1** *ExecPrefixRule Attributes*

| Attribute   | Description   |
|-------------|---|
| Delimiters  | A list of valid delimiters; you can use any character as a delimiter, such as @#-/.   |
| Prefix      | List of valid prefixes.   |
| StripPrefix | Option to strip or not to strip the prefix from the User-Name. If you configure this attribute to YES, the ExecPrefixRule strips the prefix from the User-Name. If you configure this attribute to NO, the ExecPrefixRule does not strip the prefix from the User-Name. By default, this attribute is set to YES. |

For example, if `cisco/bob@abc.com` is the User-Name attribute, the **ExecPrefixRule** script sets the Authentication-Service to `abc-service` and User-Name to:

- `bob@abc.com` when the StripPrefix attribute is set to YES.
- `cisco/bob@abc.com` when the StripPrefix attribute is set to NO.

You can configure the Prefix attribute in Prime Access Registrar using the `aregcmd` as follows:

#### set Prefix “cisco”

The Prime Access Registrar server does a case-insensitive comparison of the value specified for the prefix attribute of a username.

You can configure the Prefix by specifying a pattern using the following notation:

```
~/pattern/
```

```
[ //localhost/Radius/Rules/prefix/Attributes ]
```

```
Delimiters = #@-/
```

```
Prefix = ~/cis*/
```

Where a pattern is a string of alpha-numeric characters that can include wild card characters, as in “cis\*” to match patterns (realms) that start with “cis”.



#### Note

If you specify `/` as the delimiter while configuring ExecPrefix Rule, you must configure the prefix as **Prefix =~/pattern//**.



#### Note

The question mark (`?`) should not be used without a character pattern preceding it. Specifying `?` as the first character might have undesirable results. (For regexp terminology, the question mark should be preceded by an atom.)

## ExecSuffixRule

The Policy Engine references **ExecSuffixRule** to determine the AAA services, policy and session managers based on the suffix (or *realm*) set in the Policy Engine. You can use **aregcmd** to configure **ExecSuffixRule** to support multivalued attributes, as in the following:

```
set Suffix “cisco.com” “abc.com” “domain.com”
```

In the User-Name `bob@abc.com`, **ExecSuffixRule** first checks for any of the configured delimiters in the User-Name. If there is a match, **ExecSuffixRule** checks for the configured suffix in the User-Name. If the suffix matches, **ExecSuffixRule** checks for the value of the StripSuffix variable. If StripSuffix is set to Yes, the suffix (including the delimiter) is stripped from the User-Name attribute of the Access Request.

Table 10-2 lists the **ExecSuffixRule** script attributes.

**Table 10-2** ExecSuffixRule Attributes

| Attribute  | Description   |
|------------|---|
| Delimiters | A list of valid delimiters; you can use any character as a delimiter such as these: <code>@#/#</code> |

**Table 10-2** *ExecSuffixRule Attributes (continued)*

| Attribute   | Description  |
|-------------|--|
| Suffix      | List of valid suffixes to scan   |
| StripSuffix | The default value (No) does not strip the suffix from the User-Name. When set to Yes, <b>ExecSuffixRule</b> does strip the suffix. |

The Prime Access Registrar server does a case-insensitive comparison of the value specified for the suffix attribute for the suffix of a username.

You can also specify a pattern using the following notation:

*~/pattern/*

Where pattern is a string of alpha-numeric characters that might include wild card characters, as in “@\*cisco.com” to match patterns (realms) that end in *cisco.com*.

**Note**

The question mark (?) should not be used without a character pattern preceding it. Specifying ? as the first character might have undesirable results. (For regexp terminology, the question mark should be preceded by an *atom*.)

## Configuring Suffix and Prefix Policies

**Step 1** Activate the Policy Engine by configuring SelectPolicy.

For example, the following script explains you how to set a suffix and prefix policy in the Grouping list.

```
--> cd selectPolicy/

[ //localhost/Radius/Policies/SelectPolicy ]
  Name = SelectPolicy
  Description =
  Grouping = suffixrule&prefixrule
```

**Step 2** Run the configuration rules for Prefix and Suffix.

For example, the suffix and prefix rule configuration do the following:

- points to the **ExecSuffixRule** script
- specifies the delimiters for which to scan
- specifies the suffixes for which to scan
- indicates whether to strip the suffix from the User-Name

```
[ //localhost/Radius/Rules ]
  Entries 1 to 2 from 2 total entries
  Current filter: <all>

prefixrule/
  Name = prefixrule
  Description =
  Type = radius
  Script~ = ExecPrefixRule
  Attributes/
    Authentication-Service = local-users
```

```

Authorization-Service = local-users
Delimiters = @#%$/
Prefix = cisco
StripPrefix = no
suffixrule/
  Name = suffixrule
  Description =
  Type = radius
  Script~ = ExecRealmRule
  Attributes/
    Realm = @cisco.com

```

---

In this example, if *bob@abc.com* is the User-Name attribute, **ExecSuffixRule** strips the User-Name *bob@abc.com* and sets the User-Name environment variable to *bob* because *StripSuffix* is configured as *yes*.

## ExecTimeRule

Use the **ExecTimeRule** script to implement access restriction on users based on time. The **ExecTimeRule** script checks the time at which the request packet arrives and based on that the authorization parameters for the user can be decided. Based on the time of the request packet if the rule succeeds then **ExecTimeRule** sets the environment variable, *Acceptedprofiles* to a profile or a set of profiles.

For example, the following statement checks for *Timerange* and applies *AcceptedProfiles*.

```
Acceptedprofiles=Regularaccess::Highprivilegeaccess
```

## ParseTranslationGroupsByRealm

The Policy Engine references the *ParseTranslationGroupsByReal* script to determine the incoming and outgoing translation groups based on realm set in the Policy Engine. Use the *ParseTranslationGroupsByReal* script to add or filter attributes in request and response packets. The *ParseTranslationGroupsByReal* script supports multi-value attributes enabling you to configure to check for multiple Realms.

For instance, the following statement checks for three Realms. If True, the Policy Engine applies the values set for the *Incoming-Translation-Group* and *Outgoing-Translation-Groups* attributes.

```
set Realm "@cisco.com" "@foo.com" "@bar.com"
```

## ParseTranslationGroupsByDNIS

This script is referenced from the Policy Engine to determine the incoming and outgoing translation groups based on DNIS set in the Policy Engine. This script can be used to add/filter attributes in request/response packets. This script supports multi-value attributes, by which multiple DNIS can be configured for checking.

For example, the following statement checks for *Calling-Station-ID* and applies *Incoming-Translation-Groups* and *Outgoing-Translation-Groups*.

```
set DNIS "1111111" "2222222" "3333333"
```

## ParseTranslationGroupsByCLID

The Policy Engine references the ParseTranslationGroupsByCLID script to determine the incoming and outgoing translation groups based on CLID set in the Policy Engine. You can use the ParseTranslationGroupsByCLID script to add and filter attributes in request and response packets. The ParseTranslationGroupsByCLID script supports multi-value attributes, by which you can configure multiple CLIDs for checking.

For example, the following statement checks for the Calling-Station-ID and applies Incoming-Translation-Groups and Outgoing-Translation-Groups.

```
set CLID "1111111" "2222222" "3333333"
```

## ParseTranslationGroupsByDNIS

The ParseTranslationGroupsByDNIS script is referenced from the policy engine to determine the incoming and outgoing translation groups based on DNIS set in the policy engine. The ParseTranslationGroupsByDNIS script can be used to add and/or filter attributes in request and response packets. The ParseTranslationGroupsByDNIS script supports multi-value attributes, by which multiple DNIS can be configured for checking.

For example, the following statement checks for the Calling-Station-ID and applies Incoming-Translation-Groups and Outgoing-Translation-Groups.

```
set DNIS "1111111" "2222222" "3333333"
```





## Using FastRules to Process Packet Flow

---

While using rule policy engine and scripting points to process packet flow, you need to be familiar with programming languages, and create scripts to attach them to the Prime Access Registrar configuration. FastRules concept is an easier and efficient alternative to rule policy engine and scripting points.

FastRules provides a mechanism to easily choose the right authentication, authorization, accounting, and query service(s), drop, reject, or break flows, run a script, choose a session manager and/or a chain of fast rules required for processing a packet.

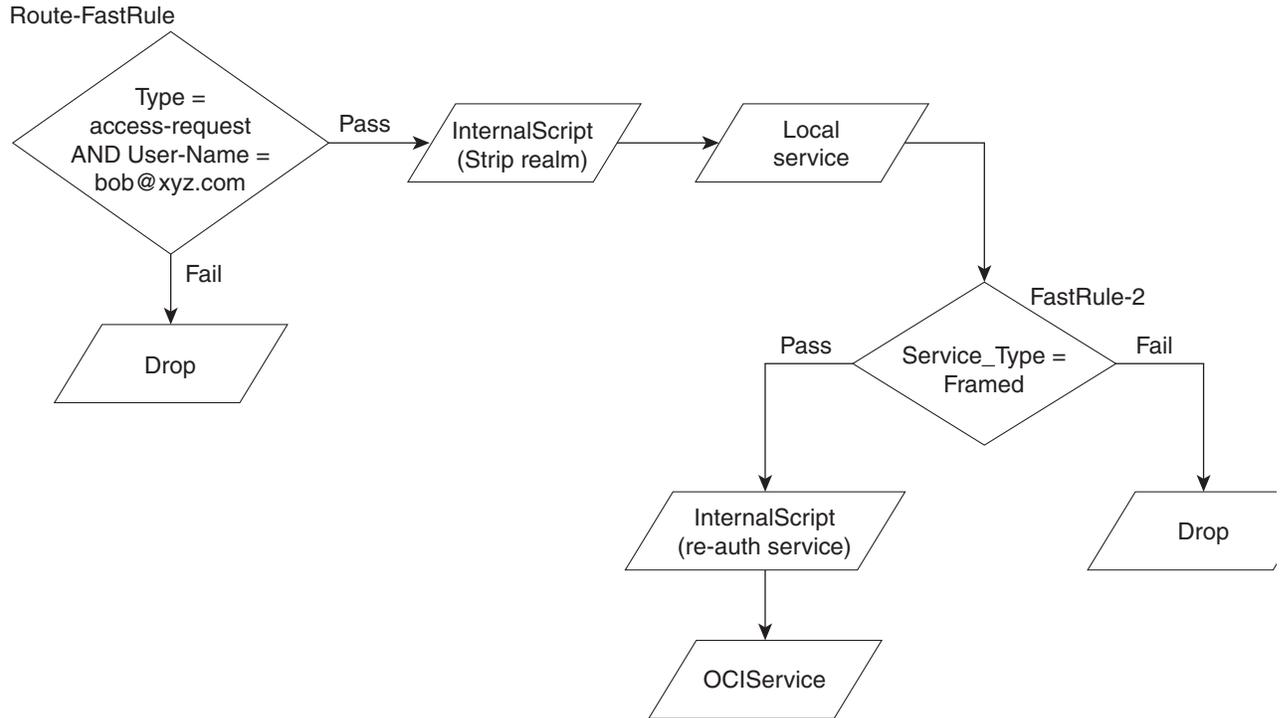
FastRules has the following capabilities:

- Provides maximum flexibility and ease in matching information in the incoming packets for choosing the appropriate service to apply
- Provides an option to match values in AVPs based on value ranges, exact match, and simple string comparisons using regex
- Provides easy and efficient alternative to rule/policy engine and scripting points for most common use cases—reduces the use of external scripts to choose an appropriate service

[Figure 11-1](#) describes the workflow for FastRules.

FastRules also provides an option to use Internal Scripts as part of the workflow. Internal scripts allow you to add, modify, or delete attributes in the request, response, and environment dictionaries for RADIUS, Diameter, and TACACS+.

Figure 11-1 FastRules Workflow



## Configuring FastRules

FastRules can be configured either through the GUI or through a CLI.

A sample CLI for configuring fast rules is given below:

```
[ //localhost/Radius/Fastrules]
  Ruledefinitions\
  radroot\
    Name = radroot
    Desc =
    Protocol = Radius
    Condition = A1 AND (A2 OR A3)
    Success = Authen(mod1)-->fr2-->Author(oci)
    Failure =
    Attributes\
    A1\
      Name = A1
      Attribute = Calling-station-id
      Value = 1.2.3.4
    A2\
      Name = A2
      Attribute = User-name
      Value = bob
    A3\
      Name = A3
      Attribute = User-name
      Value = BEGINS_WITH(jane)
  fr1\
    Name = fr1
    Desc =
```

```
Protocol = Radius
Condition = Attr1
Success = Authen(proxy)
Failure = DROP
Attributes\
Attr1\
    Name = Attr1
    Attribute = Calling-station-id
    Value = 2.3.4.5
fr2\
    Name = fr2
    Desc =
    Protocol = Radius
    Condition = Attribute1
    Success = Authen(local)
    Failure =
    Attributes\
        Attribute1\
            Name = Attribute1
            Attribute = Status
            Value = Failure
Order\
    Radius\
        1. radroot
        2. fr1
    Diameter\
        1.
    Tacacs\
        1.
Services\
    mod1\
    Name = mod1
    Desc =
    Type = mod
    Service = ldap
    Dictionary\
        +env:Calling = req:Calling-Station-Id
        -req:Calling-Station-Id = ""
        +req:Calling-Station-Id = "5.6.7.8"
        +res:Calling-Station-Id = env:Calling
        #res:Called-Station-Id = "123456789"
```





## Using LDAP

---

This chapter provides information about using Lightweight Directory Access Protocol (LDAP) with Cisco Prime Access Registrar (Prime Access Registrar) to access information directories. You can use Prime Access Registrar to authenticate and authorize access requests by querying user information through LDAP.



**Note**

---

Prime Access Registrar supports LDAP version 3 and LDAP version 2 directory servers.

---

This chapter contains the following sections:

- [Configuring LDAP](#)
- [Analyzing LDAP Trace Logs](#)
- [Bind-Based Authentication for LDAP](#)

## Configuring LDAP

To use LDAP in Prime Access Registrar, use **aregcmd** to do the following:

1. [Configuring the LDAP Service.](#)
2. [Configuring an LDAP RemoteServer.](#)
3. [Setting LDAP As Authentication and Authorization Service.](#)
4. [Saving Your Configuration.](#)

After you issue the **save** command, Prime Access Registrar attempts to validate the configuration, checks for all required properties, and ensures there is no logic error. If the validation is successful, Prime Access Registrar saves the configuration to the MCD database. When Prime Access Registrar is reloaded, it shuts down any current LDAP connections and builds new connections for the configured LDAP remote servers.

## Configuring the LDAP Service

You configure an LDAP service under **/Radius/Services**. When you define an LDAP service under **/Radius/Services**, you must set its type to LDAP.

```
[ //localhost/Radius/Services/AR-LDAP ]
  Name = AR-LDAP
  Description =
  Type = ldap
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
  OutageScript~ =
  MultipleServersPolicy = Failover
  RemoteServers/
```

Table 12-1 describes the LDAP service properties.

**Table 12-1** LDAP Service Properties

| Parameter             | Description  |
|-----------------------|--|
| Name                  | Required; inherited from the upper directory   |
| Description           | An optional description of the service   |
| Type                  | Must be set to LDAP for LDAP service   |
| IncomingScript        | Optional   |
| OutgoingScript        | Optional   |
| OutagePolicy          | Required; must be set to AcceptAll or Drop Packet, or defaults to RejectAll  |
| OutageScript          | Optional   |
| MultipleServersPolicy | Required; must be set to RoundRobin or defaults to Failover.   |
| RemoteServers         | Required; list of one or more remote servers defined under <b>/Radius/Services/LDAP/RemoteServers</b> . These servers must be listed in order under <b>/Radius/RemoteServers</b> . |

This section contains the following topics:

- [MultipleServersPolicy](#)
- [RemoteServers](#)

### MultipleServersPolicy

Use the MultipleServersPolicy property to configure the LDAP remote servers in RoundRobin mode, or the default Failover mode applies. When set to Failover, Prime Access Registrar directs requests to the first server in the **/Radius/Services/LDAP/RemoteServers** list. If that server should fail or go offline, Prime Access Registrar redirects all requests to the next server in the list. The process continues until Prime Access Registrar locates an online server.

When set to RoundRobin, Prime Access Registrar directs each request to the next server in the RemoteServers list to share the resource load across all listed servers.

## RemoteServers

Use the RemoteServers directory to list one or more remote servers to process access requests. The servers must also be listed in order under **/Radius/RemoteServers**.

The order of the RemoteServers list determines the sequence for directing access requests when MultipleServersPolicy is set to RoundRobin mode. The first server in the list receives all access requests when MultipleServersPolicy is set to Failover mode.

## Configuring an LDAP RemoteServer

Use the **aregcmd add** command to add LDAP servers under **/Radius/RemoteServers**. You must configure an LDAP RemoteServer object for each RemoteServer object you list under **/Radius/Services/LDAP/RemoteServers**.

The *Name*, *Protocol*, *Port*, *HostName*, *BindName*, *BindPassword*, *SearchPath*, and *Filter* properties must be configured to use an LDAP remote server.

The *Name*, *Protocol*, *Port*, *HostName*, *SearchPath*, and *Filter* properties must be configured to enable Bind-Based Authentication.



### Note

You can leave the BindName, BindPassword, UserPasswordAttribute, PasswordEncryptionStyle and DNSLookupAndLDAPRebindInterval properties blank when you configure the Bind-Based Authentication feature in Prime Access Registrar.

Table 12-2 describes the LDAP Remote Server properties.

**Table 12-2 LDAP Remote Server Properties**

| Parameter               | Description   |
|-------------------------|---|
| Name                    | Required name you assign  |
| Description             | Optional description of the server  |
| Protocol                | Required and must be set to LDAP; no default value  |
| Port                    | Required; port on which LDAP server listens, default is port 389.<br><b>Note</b> If port is not set or set to zero, LDAP remote server will automatically be set to port 389. |
| ReactivateTimerInterval | Required; default is 300000 (ms)  |
| Timeout                 | Required; specifies length of time Prime Access Registrar waits for a response from the LDAP server before noting the server as down; default is 15 (seconds)                 |
| HostName                | Required; specifies the hostname, FQDN, or IP address of the LDAP server  |
| BindName                | Specifies the distinguished name (DN) in the LDAP server for Prime Access Registrar to bind with the LDAP server  |
| BindPassword            | Specifies the password for the distinguished name   |
| UseSSL                  | FALSE by default  |

Table 12-2 LDAP Remote Server Properties (continued)

| Parameter                      | Description  |
|--------------------------------|--|
| SearchPath~                    | Specifies search base to the organization and domain; for example: o=cisco.com   |
| Filter~                        | (uid=%s) by default  |
| UserPasswordAttribute          | Should be set to the attribute in the directory server which stores users' passwords; default is <i>userpassword</i>   |
| LimitOutstandingRequests       | FALSE by default   |
| MaxOutstandingRequests         | Limits the number of requests to the LDAP server; used to throttle the request load when the LDAP server does not function well under high TPS rates (default is 0)  |
| MaxReferrals                   | Limits the number of referrals Prime Access Registrar allows when working with LDAPv2 (default is 0)   |
| ReferralAttribute              | LDAP attribute that contains a referral for LDAPv2   |
| ReferralFilter                 | Filter used when following a referral for LDAPv2   |
| PasswordEncryptionStyle        | <p>Dynamic by default; must be set to one of the following depending on the algorithm used by the LDAP server to encrypt passwords:</p> <ul style="list-style-type: none"> <li>Dynamic</li> <li>Crypt</li> <li>None</li> <li>SHA-1</li> <li>SSHA-1</li> </ul> <p>When set to <i>Dynamic</i>, Prime Access Registrar analyzes the password and detects the encryption algorithm used.</p> <p><i>None</i> indicates that the LDAP server stores clear text passwords.</p> <p><b>Note</b> If CHAP authentication is used with LDAP backing store, passwords in LDAP must be stored as clear text.</p> |
| EscapeSpecialCharInUserName    | FALSE by default   |
| DNSLookupAndLDAPRebindInterval | Specifies the timeout period after which the Prime Access Registrar server will attempt to resolve the LDAP hostname to IP address (DNS resolution); 0 by default  |
| DataSourceConnections          | Specifies the number of concurrent connections to the LDAP server. The default value is 8.   |
| SearchScope                    | <p>Specifies how deep to search within a search path; default is <i>SubTree</i> which indicates a search of the base object and the entire subtree of which the base object distinguished name is the highest object.</p> <p><i>Base</i> indicates a search of the base object only.</p> <p><i>OneLevel</i> indicates a search of objects immediately subordinate to the base object, but does not include the base object.</p>  |

Table 12-2 LDAP Remote Server Properties (continued)

| Parameter                   | Description  |
|-----------------------------|--|
| LDAPToRadiusMappings        | <p>Optional; a list of name/value pairs in which the name is the name of the <b>ldap</b> attribute to retrieve from the user record, and the value is the name of the RADIUS attribute to set to the value of the <b>ldap</b> attribute retrieved.</p> <p>For example, when the <b>LDAPToRadiusMappings</b> has the entry: <b>FramedIPAddress = Framed-IP-Address</b>, the RemoteServer retrieves the <b>FramedIPAddress</b> attribute from the <b>ldap</b> user entry for the specified user, uses the value returned, and sets the Response variable <b>Framed-IP-Address</b> to that value.</p> |
| LDAPToEnvironmentMappings   | <p>Optional; a list of name/value pairs in which the name is the name of the <b>ldap</b> attribute to retrieve from the user record, and the value is the name of the Environment variable to set to the value of the <b>ldap</b> attribute retrieved.</p> <p>For example, when the <b>LDAPToEnvironmentMappings</b> has the entry: <b>group = User-Group</b>, the RemoteServer retrieves the <b>group</b> attribute from the <b>ldap</b> user entry for the specified user, uses the value returned, and sets the Environment variable <b>User-Group</b> to that value.</p>                       |
| LDAPToCheckItemMappings     | <p>Optional; a list of LDAP <i>attribute/value</i> pairs which must be present in the RADIUS access request and must match, both name and value, for the check to pass.</p> <p>For example, when the <b>LDAPToCheckItemMappings</b> has the entry: <b>group = User-Group</b>, the Access Request must contain the attribute <b>group</b>, and it must be set to <b>User-Group</b>.</p>   |
| UseBindBasedAuthentication  | <p>A boolean field that enables bind-based authentication with LDAP server. By default, this property is set to FALSE that uses existing legacy authentication method.</p> <p>On setting this property to TRUE, the mappings LDAPToRadius, LDAPToEnvironment, and LDAPToCheckItem will not work.</p>   |
| UseBinaryPasswordComparison | <p>A boolean value that enables binary-based password comparison to authenticate. This property when set to TRUE, enables binary password comparison. By default this property is set to FALSE.</p>  |

This section contains the following topics:

- [DNS Look Up and LDAP Rebind Interval](#)
- [LDAPToRadiusMappings](#)
- [LDAPToEnvironmentMappings](#)
- [LDAPToCheckItemMappings](#)

## DNS Look Up and LDAP Rebind Interval

Prime Access Registrar provides a DNS Look-up and LDAP Rebind feature that enables you to use a smart DNS server for LDAP hostname resolution, allows you to query a DNS server at set intervals to resolve the LDAP hostname, and optionally rebind to the LDAP server, if necessary.

When you configure Prime Access Registrar to use an LDAP directory server, you can specify the hostname of the LDAP directory server. The hostname can be a qualified or an unqualified name. You can also specify a timeout period after which Prime Access Registrar will again resolve the hostname. If the IP address returned is different from the previous, Prime Access Registrar establishes a new LDAP bind connection.

The `DNSLookupAndLDAPRebindInterval` property specifies the timeout period after which the Prime Access Registrar server will attempt to resolve the LDAP hostname to IP address (DNS resolution). When you do not modify `DNSLookupAndLDAPRebindInterval`, the default value zero indicates the server will perform normal connection and binding only at start-up time or during a reload. Unless you change the default to a value greater than zero, the server will not perform periodic DNS lookups.

Prime Access Registrar maintains and uses the existing bind connection until a new one is established to minimize any performance impact during the transfer. Prime Access Registrar ensures that no requests are dropped or lost during the transfer to a new LDAP binding.

Set the `DNSLookupAndLDAPRebindInterval` using a numerical value and the letter H for hours or M for minutes, such as in the following examples:

```
set DNSLookupAndLDAPRebindInterval 15M—performs DNS resolution every 15 minutes
```



### Note

We recommend that you do not set `DNSLookupAndLDAPRebindInterval` to a value less than 15 minutes to minimize its effect on server performance.

```
set DNSLookupAndLDAPRebindInterval 1h—performs DNS resolution every hour
```

### Configure the DNS Look-up and LDAP Rebind

To configure the DNS Look-up and LDAP Rebind,

- 
- Step 1** Log into the Prime Access Registrar server, and use `aregcmd` to navigate to `//localhost/Radius/Remoteservers`. If necessary, add the LDAP server, or change directory to it.

```
cd /Radius/RemoteServers/ldap-serv1/
```

- Step 2** Set the `DNSLookupAndLDAPRebindInterval` property to the interval time desired.

```
set DNSLookupAndLDAPRebindInterval 30 M
```

---

## LDAP Rebind Failures

Prime Access Registrar records any name resolution failures, bind successes and failures, and the destination hostname and IP address in the log file. At trace level 3, Prime Access Registrar also logs the time of any new bind connections and the closing of any old bind connections.

If either the name resolution or bind attempt fail, Prime Access Registrar continues using the existing bind connection until the timeout has expired again. If there is no existing bind connection, Prime Access Registrar marks the remote server object as *down*.

## LDAPToRadiusMappings

Configure LDAPToRadiusMappings with a list of *name/value* pairs where name is the name of the data store attribute to retrieve from the user record and the value is the name of the RADIUS attribute to set to the value of the data store attribute retrieved.

Values stored in a multivalued field in the LDAP directory are mapped to multiple RADIUS attributes. For example, if the LDAPToRadiusMappings has the following entry:

```
tunnel-info = Cisco-AVPair
```

The following LDAP fields in the user's record will create four Cisco-AVPair attributes in the user's Access-Accept RADIUS packet:

```
tunnel-info: vpdn:tunnel-id=ssg001
tunnel-info: vpdn:tunnel-type=12tp
tunnel-info: vpdn:ip-addresses=10.2.2.2
tunnel-info: vpdn:12tp-tunnel-password=secret
```

## LDAPToEnvironmentMappings

LDAPToEnvironmentMappings comprises a list of attribute name/value pairs or AV pairs where the name is the name of the data store attribute to retrieve from the user record, and the value is the name of the Environment variable to set to the value of the LDAP attribute retrieved.

For example, when the LDAPToEnvironmentMappings has the entry: `group=User-Group`, the RemoteServer retrieves the attribute from the LDAP user entry for the specified user, uses the value returned, and sets the Environment variable User-Group to that value.

## LDAPToCheckItemMappings

LDAPToCheckItemMappings comprises a list of LDAP AV pairs which must be present in the RADIUS access request and must match, both name and value, for the check to pass. Prime Access Registrar will first authenticate the user's password in the Access-Request before validating the check item attributes.

## Setting LDAP As Authentication and Authorization Service

Use **aregcmd** to configure the LDAP Service as the default authentication and authorization service under **/Radius** as in the following:

```
set DefaultAuthenticationService AR-LDAP
```

```
set DefaultAuthorizationService AR-LDAP
```

## Saving Your Configuration

When you use **aregcmd** to **save** your configuration, Prime Access Registrar does the following:

- Attempts to validate the configuration

- Checks for all required parameters
- Ensures there are no logic errors

If the validation is successful, Prime Access Registrar saves the configuration to the MCD database. When you **reload**, Prime Access Registrar shuts down any current LDAP connections and builds new connections for the configured LDAP servers.

This section contains the following topics:

- [CHAP Interoperability with LDAP](#)
- [Allowing Special Characters in LDAP Usernames](#)
- [Dynamic LDAP Search Base](#)

## CHAP Interoperability with LDAP

If the you plan to use CHAP authentication with an LDAP backing store, the password in LDAP must be stored as clear text. This is due to the one-way hash used by the CHAP, crypt, SHA-1, and SSHA encryption algorithms.

## Allowing Special Characters in LDAP Usernames

This feature allows you to use special characters in LDAP usernames. The allowable special characters are \*, (, and \. These special characters can be included in the string passed to LDAP as the LDAP username value (usually the RADIUS username attribute).

The default of `EscapeSpecialCharInUserName` is `FALSE`. To enable this feature, use **aregcmd** to set the `EscapeSpecialCharInUserName` attribute in `/Radius/RemoteServers/ldap-server` to `TRUE`, as shown in the following example.

```
cd /Radius/RemoteServers/ldap-server

set EscapeSpecialCharInUserName TRUE

/Radius/RemoteServers/Ldap-Server
EscapeSpecialCharInUserName = TRUE
```



### Note

---

This feature supports the LDAP V3 library.

---

## Dynamic LDAP Search Base

A new environment variable, `Dynamic-Search-Path` (see **rex.h**), can be used to set the dynamic LDAP search base. If this environment variable is defined for an LDAP service, it will override the default LDAP search base defined in the LDAP Remote Server configuration. This allows the LDAP search base to be configured on a per-user basis.

For example, you could match the search base to the organization and domain (in a Tcl script called from `/Radius/IncomingScript`):

```
set user [ $request get User-Name ]
if { [ regexp {^[^@]+@[^\.\.]+\.(.+)$} $user m org domain ] } {
$environ put Dynamic-Search-Path "ou=$org,ou=people,o=$domain"
```

# Analyzing LDAP Trace Logs

Prime Access Registrar records in the log files any name resolution failures, bind successes and failures, and the destination hostname and IP address. At trace level 3, Prime Access Registrar logs the time of any new bind connections and the closure of any old bind connections and also information about user login requests and reply messages.

This section contains the following topics:

- [Successful Bind Message](#)
- [Bind Failure Messages](#)
- [Login Failure Messages](#)

## Successful Bind Message

The following message is logged in the **name\_radius\_1\_trace** file, when the Prime Access Registrar server successfully binds to the LDAP server. In this case, spatula-u5 is the LDAP server listening on port number 389.

```
10/12/2013 11:02:57: Log: Successfully bind to LDAP Server ldapserver (spatula-u5:389)
```

## Bind Failure Messages

The following messages are logged in the **name\_radius\_1\_trace** file, when Prime Access Registrar server fails to bind to the LDAP server.

```
10/12/2013 11:10:50: Log: Write in LDAPClient returned an error (32)
```

```
10/12/2013 11:10:50: Log: Remote LDAP Server ldapserver (spatula-u5:387): Unable to bind to LDAP Server: Can't contact LDAP server
```

```
10/12/2013 11:10:50: Log: Remote LDAP Server ldapserver (spatula-u5:387): Failed to open the connection to the LDAP server
```

Messages like those above could indicate that the hostname specified does not resolve to the correct IP address of the LDAP server or the configured port number might not be the port on which the LDAP server listens.

The following messages are logged in the **name\_radius\_1\_trace** file, when Prime Access Registrar server fails to bind to the LDAP server.

```
10/12/2013 11:45:14: Log: Remote LDAP Server ldapserver (spatula-u5:389): Unable to bind to LDAP Server: No such object ()
```

```
10/12/2013 11:45:14: Log: Remote LDAP Server ldapserver (spatula-u5:389): Failed to open the connection to the LDAP server
```

The Distinguished Name (DN) provided in the BindName property was invalid. The DN provided in the BindName property should contain the exact string used in the directory server to define the object.

The following messages are logged in the **name\_radius\_1\_trace** file, when Prime Access Registrar server fails to bind to the LDAP server.

```

10/12/2013 11:51:55: Log: Remote LDAP Server ldapserver (spatula-u5:389): Unable to
bind to LDAP Server: Invalid credentials
10/12/2013 11:51:55: Log: Remote LDAP Server ldapserver (spatula-u5:389): Failed to
open the connection to the LDAP server

```

The messages above indicate that the password provided in the BindPassword property was incorrect.

## Login Failure Messages

The following messages are logged in the **name\_radius\_1\_trace** file, when user *jane* tries to login. These messages indicate that user *jane* does not have a record in the directory server or the SearchPath property has an incorrect value. The SearchPath property should have the directory where the user record is stored in the directory server.

Notice how the messages specify the service, remote LDAP server, username, and contents of the Access-Reject packet.

```

10/12/2013 11:24:17: P8457: Authenticating and Authorizing with Service AR-LDAP
10/12/2013 11:24:17: id = 5
10/12/2013 11:24:17: P8457: Remote LDAP Server ldapserver (spatula-u5: 389): Querying
LDAP server, id = 5.
10/12/2013 11:24:17: P8457: Remote LDAP Server ldapserver (spatula-u5: 389): GotLDAP
response, id = 5.
10/12/2013 11:24:17: P8457: Remote LDAP Server ldapserver (spatula-u5: 389): No
matching entries returned from LDAP query.
10/12/2013 11:24:17: P8457: User jane was not found in the LDAP store
10/12/2013 11:24:17: P8457: Rejecting request
10/12/2013 11:24:17: P8457: Rejecting request
10/12/2013 11:24:17: P8457: Trace of Access-Reject packet
10/12/2013 11:24:17: P8457: identifier = 4
10/12/2013 11:24:17: P8457: length = 35
10/12/2013 11:24:17: P8457: reqauth = 01:ad:cf:c7:4f:8e:a4:38:b0:d8:0a:e5:3d:9f:64:16
10/12/2013 11:24:17: P8457: Reply-Message = Access Denied

```

The following messages are logged in the **name\_radius\_1\_trace** file, when user *bob* tries to login. These messages indicate that user *bob* tried to login with an incorrect password.

```

10/12/2013 11:36:59: P8461: Authenticating and Authorizing with Service AR-LDAP
10/12/2013 11:36:59: id = 7
10/12/2013 11:36:59: P8461: Remote LDAP Server ldapserver (spatula-u5: 389): Querying
LDAP server, id = 7.
10/12/2013 11:36:59: P8461: Remote LDAP Server ldapserver (spatula-u5: 389): Got LDAP
response, id = 7.
10/12/2013 11:36:59: P8461: Remote Server ldapserver (spatula-u5:389): User bob's
password does not match
10/12/2013 11:36:59: P8461: User bob's password does not match
10/12/2013 11:36:59: P8461: Rejecting request
10/12/2013 11:36:59: P8461: Rejecting request
10/12/2013 11:36:59: P8461: Trace of Access-Reject packet
10/12/2013 11:36:59: P8461: identifier = 6
10/12/2013 11:36:59: P8461: length = 35
10/12/2013 11:36:59: P8461: reqauth = de:8d:4b:c4:f9:c0:06:a6:98:2d:8c:e9:f3:a9:a3:c2
10/12/2013 11:36:59: P8461: Reply-Message = Access Denied

```

The following messages are logged in the **name\_radius\_1\_trace** file, when user *bob* tries to login. These messages indicate the user record for user *bob* does not contain an attribute called pass. The UserPasswordAttribute property has an incorrect value called *pass*. The UserPasswordAttribute property should have the attribute name in the directory records where the user password is stored.

```

10/12/2013 12:02:09: P9865: Authenticating and Authorizing with Service AR-LDAP
10/12/2013 12:02:09: id = 2

```

```

10/12/2013 12:02:09: P9865: Remote LDAP Server ldapserver (spatula-u5: 389): Querying
LDAP server, id = 2.
10/12/2013 12:02:09: P9865: Remote LDAP Server ldapserver (spatula-u5: 389): Got LDAP
response, id = 2.
10/12/2013 12:02:09: P9865: Remote LDAP Server ldapserver (spatula-u5: 389): LDAP
entry for user bob did not have a password (" pass") attribute
10/12/2013 12:02:09: P9865: User bob's password does not match
10/12/2013 12:02:09: P9865: Rejecting request
10/12/2013 12:02:09: P9865: Rejecting request
10/12/2013 12:02:09: P9865: Trace of Access-Reject packet
10/12/2013 12:02:09: P9865: identifier = 10
10/12/2013 12:02:09: P9865: length = 35
10/12/2013 12:02:09: P9865: reqauth = 0d:b6:83:f9:e8:3d:a4:ad:f1:c9:33:72:91:0b:29:1c
10/12/2013 12:02:09: P9865: Reply-Message = Access Denied

```

**Note**

Remember to **reload** the Prime Access Registrar server after any changes to the LDAP server configuration.

## Bind-Based Authentication for LDAP

Prime Access Registrar supports most of the LDAP servers. But, a few of the LDAP servers do not support the functionality of Prime Access Registrar, which gets the passwords from the LDAP and matches them in Prime Access Registrar.

The bind-based authentication feature in Prime Access Registrar allows you to use any LDAP server; it verifies the password in the LDAP database instead of the Prime Access Registrar database. When Prime Access Registrar receives a request, it sends the username and password to the LDAP server. The LDAP server searches for a match, and approves the request if it finds a matching user credential in its database. It rejects the request if it does not find any matching credentials.

### Configuring Bind-Based Authentication for LDAP

To configure the bind-based authentication for LDAP,

**Step 1** Launch **aregcmd**.

**Step 2** Create an **LDAP** service.

```
[ //localhost/Radius ]
```

```
cd Services/
```

```
add ldap
```

```
cd ldap
```

```
set Type ldap
```

```
[ //localhost/Radius/Services/ldap ]
```

```

Name = ldap
Description =
Type = ldap
IncomingScript~ =
OutgoingScript~ =
OutagePolicy~ = RejectAll
OutageScript~ =
MultipleServersPolicy = Failover
RemoteServers/

```

```
cd RemoteServers
```

```
add 1 ldapserver
```

**Step 3** Create the LDAP Remote Server Object.

```
[ //localhost/Radius ]
```

```
cd RemoteServers
```

```
add ldapserver
```

```
cd ldapserver
```

```
[ //localhost/Radius/RemoteServers/ldap ]
```

```
set Port <remote ldap server prt numer>
```

```
set HostName <remote ldap server name/ipaddress>
```

```
set SearchPath <configured in ldap server>
```

```
set UseBindBasedAuthentication TRUE
```

```
cd /Radius
```

```
set DefaultAuthenticationService <ldap service name>
```

```
set DefaultAuthorizationService <ldap service name>
```

**Step 4** Save the configuration.

```
save
```

**Step 5** Restart the application.

```
reload
```

---







## Using Open Database Connectivity

---

Cisco Prime Access Registrar (Prime Access Registrar) supports Open Database Connectivity (ODBC) and Oracle Call Interface (OCI), open specifications that provide application developers a vendor-independent API with which to access data sources. For ODBC, Prime Access Registrar supports MySQL database connectivity and for OCI, it supports Oracle database connectivity. It provides RemoteServer objects and services to support ODBC or OCI. You can use Prime Access Registrar to authenticate and authorize access requests by querying user information through ODBC or OCI.

ODBC or OCI is an application program interface (API). Real data exchange between an application and data store is still carried out by SQL through ODBC or OCI. To achieve the most flexibility, you are required to define your own SQL using **aregcmd**. Prime Access Registrar will register the SQL statements and send them to the data store through ODBC or OCI when required. Because you can define your own SQL, Prime Access Registrar supports sites that have their own data stores.

ODBC is configured using **.ini** files, specifically **odbc.ini** and **odbcinst.ini**. However, you cannot create or modify these files directly. Prime Access Registrar creates the **.ini** files after you use **aregcmd** to configure the ODBC connection. The SQL is stored in the local database (MCD). During execution, the Prime Access Registrar server reads the local database, prepares the SQL statements, and sends the SQL to the data source.



**Note**

---

For OCI, the **.ini** files are not needed to connect to the database.

---



**Note**

---

Prime Access Registrar uses its own ODBC driver manager and does not share existing ODBC drivers (if you already have ODBC installed). If you are already using ODBC, you will have to maintain two separate ODBC installations.

---

The ODBC or OCI memory requirement depends on your configuration. The more datasources you configure, the more memory is required. Packet processing time might increase if you configure a large number of SQL statements under SQLDefinition.

The Prime Access Registrar package includes some ODBC and OCILib Drivers, and you should use the included driver whenever possible. If a data store's ODBC driver is not included with Prime Access Registrar, you are required to install it. You configure the driver library using **aregcmd** to modify the associated **ini** file.

This chapter contains the following sections:

- [Oracle Software Requirements](#)
- [Configuring ODBC/OCI](#)
- [MySQL Support](#)

## Oracle Software Requirements

The Prime Access Registrar ODBC feature requires that you have MySQL driver packages. The OCI feature requires that you have Oracle client software installed. Supported Oracle client versions are 10.2.0.1.0 - 12c. All Oracle client software library files are expected under `$ORACLE_HOME/lib`.

When you install Prime Access Registrar software, the installation process prompts you for `ORACLE_HOME` variable and sets it in the Prime Access Registrar start-up script, `/etc/init.d/arserver`. Two other environment variables (`ODBCINI` and `ODBCSYSINI`) are also set in the `arserver` script. To change any of these variables, modify the `/etc/init.d/arserver` script and restart the Prime Access Registrar server.



### Note

For OCI services, ensure that you have installed the Oracle client properly by using `tnsping` or `sqlplus` utilities. Oracle Instant Client libraries are not supported by OCI services.

## Configuring ODBC/OCI

You use `aregcmd` to define your ODBC configuration and SQL statements. The Prime Access Registrar server automatically creates the `ODBC.ini` file for your driver manager and driver based on how you configure ODBC.

### Configuring the ODBC and ODBC-Accounting Remote Servers

To use ODBC in Prime Access Registrar for AA:

- 
- Step 1** Configure an ODBC DataSource.
  - Step 2** Configure an ODBC RemoteServer object with protocol type as 'odbc'.
  - Step 3** Configure an ODBC Service with service type as 'odbc'.
  - Step 4** Set ODBC service as the DefaultAuthenticationService and DefaultAuthorizationService.
  - Step 5** Save your configuration.
- 

To use ODBC in Prime Access Registrar for Accounting:

- 
- Step 1** Configure an ODBC DataSource.
  - Step 2** Configure an ODBC RemoteServer object with protocol type as 'odbc-account'.
  - Step 3** Configure an ODBC Service with service type as 'odbc-accounting'.
  - Step 4** Set ODBC service as the DefaultAccountingService.

**Step 5** Save your configuration.

After you **save** and validate your configuration, it is saved in the MCD database. If you have configured an ODBC service, Prime Access Registrar will query the MCD database and create or modify the **odbc.ini** file before it builds a connection to the database. When you reload your configuration, Prime Access Registrar shuts down any existing ODBC connections, then queries the MCD database to create or modify the **odbc.ini** file and build a new connection for any configured ODBC Data Sources.

The following shows an example configuration for AA remote server:

```
[ //localhost/Radius/RemoteServers/oracle-access ]
  Name = oracle-access
  Description =
  Protocol = odbc
  ReactivateTimerInterval = 300000
  Timeout = 15
  DataSourceConnections = 8
  ODBCDataSource = gordon
  SNMPTrapIP =
  SNMPTrapPort = 1521
  KeepAliveTimerInterval = 0
  SQLDefinition/
  UserPasswordAttribute = password
  SQLStatements/
  Entries 1 to 1 from 1 total entries
  Current filter: <all>
  sql1/
    Name = sql1
    Description =
    Type = query
    SQL = "select password , username from arusers where username = ?"
    ExecutionSequenceNumber = 1
    MarkerList = UserName/SQL_CHAR
  ODBCToRadiusMappings/
  ODBCToEnvironmentMappings/
  ODBCToCheckItemMappings/
```

The following shows an example configuration for AAA remote server:

```
[ //localhost/Radius/RemoteServers/ora_acc ]
  Name = ora_acc
  Description =
  Protocol = odbc-accounting
  ReactivateTimerInterval = 1000
  Timeout = 15
  DataSourceConnections = 8
  ODBCDataSource = gordon
  SNMPTrapIP =
  SNMPTrapPort = 1521
  KeepAliveTimerInterval = 1000
  BufferAccountingPackets = TRUE
  MaximumBufferFileSize = "10 Megabytes"
  NumberOfRetriesForBufferedPacket = 3
  BackingStoreEnvironmentVariables =
  UseLocalTimeZone = FALSE
  AttributeList =
  Delimiter =
  SQLDefinition/
  SQLStatements/
  Entries 1 to 1 from 1 total entries
  Current filter: <all>
  sql/
```

```
Name = sql
Description =
Type = insert
SQL = "insert into accounting(username,acct_status_type) values ( ? , ? )"
ExecutionSequenceNumber = 1
MarkerList = "UserName/SQL_CHAR Acct-Status-Type/SQL_CHAR "
```

You use **aregcmd** to define your OCI configuration and SQL statements.

### Configuring an OCI and OCI-Accounting Remote Servers

To use OCI in Prime Access Registrar for AA:

- 
- Step 1** Configure the DataSource type as oracle\_oci.
  - Step 2** Configure an OCI RemoteServer object protocol type as 'oci'.
  - Step 3** Configure an OCI Service with type as 'oci'.
  - Step 4** Set OCI service as the DefaultAuthenticationService and DefaultAuthorizationService.
  - Step 5** Save your configuration.
- 

To use OCI in Prime Access Registrar for Accounting:

- 
- Step 1** Configure the DataSource type as oracle\_oci.
  - Step 2** Configure an OCI RemoteServer object protocol type as 'oci-accounting'.
  - Step 3** Configure an OCI Service with type as 'oci-accounting'.
  - Step 4** Set OCI service as the DefaultAccountingService.
  - Step 5** Save your configuration.
- 

After you **save** and validate your configuration, it is saved in the MCD database.

The following shows an example configuration for OCI AA remote server:

```
[ //localhost/Radius/RemoteServers/AA_OCI]

Name = AA_OCI
Description =
Protocol = oci
ReactivateTimerInterval = 60000
Timeout = 1
OCITimeOutCount = 2
OCIActiveConnectionThresholdCount = 2
OCIConnectionReactivationInterval = 30000
DataSourceConnections = 4
ODBCDataSource = VM011DB
SNMPTrapIP =
SNMPTrapPort = 1521
KeepAliveTimerInterval = 3000
SQLDefinition/
  UserPasswordAttribute = password
  SQLStatements/
    Entries 1 to 1 from 1 total entries
    Current filter: <all>
```

```

sql1/
  Name = sql1
  Description =
  Type = procedure
  SQL = "call Access_Request(?,?)"
  ExecutionSequenceNumber = 1
  MarkerList = "UserName/SQL_CHAR password/SQL_OUT"
OCIToRadiusMappings/
OCIToEnvironmentMappings/
OCIToCheckItemMappings/

```

The following shows an example configuration for OCI AAA remote server:

```

[ //localhost/Radius/RemoteServers/oracle-accounting ]

Name = accounting
Description =
Protocol = oci-accounting
ReactivateTimerInterval = 6000
Timeout = 3
OCITimeOutCount = 2
OCIActiveConnectionThresholdCount = 2
OCIConnectionReactivationInterval = 3000
DataSourceConnections = 4
ODBCDataSource = VM011DB
SNMPTrapIP =
SNMPTrapPort = 1521
KeepAliveTimerInterval = 0
BufferAccountingPackets = FALSE
MaximumBufferFileSize = "10 Megabytes"
NumberOfRetriesForBufferedPacket = 3
BackingStoreEnvironmentVariables =
UseLocalTimeZone = FALSE
OCIAutoCommit = TRUE
OCITransactionCount = 0
AttributeList =
Delimiter =
SQLDefinition/
  SQLStatements/
    Entries 1 to 1 from 1 total entries
    Current filter: <all>

sql/
  Name = sql
  Description =
  Type = procedure
  SQL = "call Accounting_Request(?,?)"
  ExecutionSequenceNumber = 1
  MarkerList = "UserName/SQL_CHAR Acct-Status-Type/SQL_CHAR"

```

This section contains the following topics:

- [Configuring an ODBC/OCI Service](#)
- [Configuring an ODBC/OCI RemoteServer](#)
- [Configuring an ODBC DataSource](#)
- [Setting ODBC/OCI As Authentication and Authorization Service](#)
- [Setting ODBC/OCI As Accounting Service](#)
- [Saving Your Configuration](#)
- [Oracle Stored Procedures](#)

## Configuring an ODBC/OCI Service

You configure an ODBC or OCI service under **/Radius/Services**. When you define an ODBC or OCI service under **/Radius/Services**, you must set its type to ODBC or OCI and provide the following configuration options:


**Note**

We will use ODBC or OCI as the ODBC or OCI service name in the following examples.

### Example configuration for ODBC

```
[ //localhost/Radius/Services/ODBC ]
  Name = ODBC
  Description =
  Type = odbc
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
  OutageScript~ =
  MultipleServersPolicy = Failover
  RemoteServers/
```

### Example configuration for OCI

```
[ //localhost/Radius/Services/OCI ]
  Name = OCI
  Description =
  Type = oci
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
  OutageScript~ =
  MultipleServersPolicy = Failover
  RemoteServers/
```

[Table 13-1](#) describes the ODBC or OCI service parameters.

**Table 13-1** ODBC/OCI Service Parameters

| Parameter      | Description   |
|----------------|---|
| Name           | Required; inherited from the upper directory                                |
| Description    | An optional description of the service                                      |
| Type           | Must be set to ODBC for ODBC service or OCI for OCI service                 |
| IncomingScript | Optional  |
| OutgoingScript | Optional  |
| OutagePolicy   | Required; must be set to AcceptAll or Drop Packet, or defaults to RejectAll |
| OutageScript   | Optional  |

**Table 13-1** ODBC/OCI Service Parameters (continued)

| Parameter             | Description  |
|-----------------------|--|
| MultipleServersPolicy | Required; must be set to RoundRobin or defaults to Failover.<br><br>When set to Failover, Prime Access Registrar directs requests to the first server in the list until it determines the server is offline. If so, Prime Access Registrar redirects all requests to the next server in the list until it finds an online server.<br><br>When set to RoundRobin, Prime Access Registrar directs each request to the next server in the RemoteServers list to share the resource load across all servers in the RemoteServers list. |
| RemoteServers         | Required list of remote servers defined under <b>/Radius/Services/ODBC/RemoteServers</b> such as <b>ODBC-Primary</b> and <b>ODBC-Secondary</b>   |

## Configuring an ODBC/OCI RemoteServer

### Configuring an ODBC Remote Server

You must configure an ODBC RemoteServer object for each RemoteServer object you list under **/Radius/Services/ODBC/RemoteServers**. Use the **aregcmd** command **add** to add ODBC servers under **/Radius/RemoteServers**.

### Configuring an OCI Remote Server

You must configure an OCI RemoteServer object for each RemoteServer object you list under **/Radius/Services/OCI/RemoteServers**. Use the **aregcmd** command **add** to add OCI servers under **/Radius/RemoteServers**.

[Table 13-2](#) describes the ODBC or OCI service parameters. The fields that are displayed in the table changes based on the protocol type selected.

**Table 13-2** ODBC/OCI Remote Server Parameters

| Parameter                          | Description  |
|------------------------------------|--|
| Name                               | Required; inherited from the upper directory   |
| Description                        | An optional description of the server  |
| Protocol                           | Required and must be set to ODBC or OCI for ODBC or OCI service respectively; no default value   |
| ReactivateTimerInterval            | Required; default is 300000 (ms)   |
| Timeout                            | Required; default is 15 (seconds)  |
| OCITimeOutCount                    | Required; continuous timeout count to disconnect the selected connection. Default is 10.   |
| OCIActiveConnectionThreshold Count | Required; threshold count of disconnections after which Prime Access Registrar will mark the remote server as down and try to reactivate it. Default value is 4. |
| OCIConnectionReactivationInterval  | Required; time interval for attempting to reconnect the disconnected OCI remote server session. Default value is 3000 ms.  |

Table 13-2 ODBC/OCI Remote Server Parameters (continued)

| Parameter                                     | Description  |
|---|--|
| DataSourceConnections                         | Required; number of concurrent connections to data source (default is 8)   |
| ODBCDataSource                                | Required; no default value   |
| SQLDefinition                                 | SQLDefinition/ (mandatory, no default);<br>UserPasswordAttribute = (mandatory, no default; data store field for user password)<br>SQLStatements/<br>SQLStatement1/<br>SQLStatement2/   |
| ODBCToRadiusMappings<br>(OCIToRadiusMappings) | Optional; a list of name/value pairs in which the name is the name of the <b>odbc</b> attribute to retrieve from the user record, and the value is the name of the RADIUS attribute to set to the value of the <b>odbc</b> attribute retrieved.<br><br>For example, when the <b>ODBCToRadiusMappings</b> has the entry: <b>FramedIPAddress = Framed-IP-Address</b> , the RemoteServer retrieves the <b>FramedIPAddress</b> attribute from the <b>odbc</b> user entry for the specified user, uses the value returned, and sets the Response variable <b>Framed-IP-Address</b> to that value. |
|   |  <p><b>Note</b> When you select the protocol as OCI, the field name will be displayed as OCIToRadiusMappings.</p>   |

Table 13-2 ODBC/OCI Remote Server Parameters (continued)

| Parameter   | Description  |
|---|--|
| ODBCToEnvironmentMappings<br>(OCIToEnvironmentMappings) | <p>Optional; a list of name/value pairs in which the name is the name of the <b>odbc</b> attribute to retrieve from the user record, and the value is the name of the Environment variable to set to the value of the <b>odbc</b> attribute retrieved.</p> <p>For example, when the <b>ODBCToEnvironmentMappings</b> has the entry: <b>group = User-Group</b>, the RemoteServer retrieves the <b>group</b> attribute from the <b>odbc</b> user entry for the specified user, uses the value returned, and sets the Environment variable <b>User-Group</b> to that value.</p> <p> <b>Note</b> When you select the protocol as OCI, the field name will be displayed as OCIToEnvironmentMappings.</p> |
| ODBCToCheckItemMappings<br>(OCIToCheckItemMappings)     | <p>Optional; a list of ODBC <i>attribute/value</i> pairs which must be present in the RADIUS access request and must match, both name and value, for the check to pass.</p> <p>For example, when the <b>ODBCToCheckItemMappings</b> has the entry: <b>group = User-Group</b>, the Access Request must contain the attribute <b>group</b>, and it must be set to <b>User-Group</b>.</p> <p> <b>Note</b> When you select the protocol as OCI, the field name will be displayed as OCIToCheckItemMappings.</p>  |

## OCI Connection Timeout and Disconnection

Any single connection from Prime Access Registrar to Oracle server will be disconnected when one of the following is observed:

- Occurrence of native Oracle errors that are configured under **AdditionalNativeOracleErrors**.
- Continuous query timeouts (configured). This is configured using the **OCITimeOutCount** parameter.

This single connection disconnect will not impact the other active connections to that remote server. Hence, this will hold the state of the remote server in Prime Access Registrar as active.

Once a connection disconnects, it will attempt to reconnect after a reactivation time interval. You can configure this interval with the **OCIConnectionReactivationInterval** parameter.

Any Oracle server that Prime Access Registrar connects to will be marked as down during one of the following circumstances:

- Total number of disconnections reaches a threshold value. You can configure this threshold value using the **OCIActiveConnectionThresholdCount** parameter.
- Configured application times out—timeout in server/queue reaches the configured timeout (timeout count X number of connections).
- When the remote server starts or reactivates, no active connections are available even after waiting for the configured initial timeout.

In all the above cases, the Prime Access Registrar will attempt to re-establish the remote server connection after reactivation timer expires.

## ODBC Data Source

ODBCDataSource is the name of the datasource to be used by the remote server. An ODBCDataSource name can be reused by multiple remote servers. You configure ODBCDataSources under **/Radius/Advanced/ODBCDataSources**. See [Configuring an ODBC DataSource, page 13-13](#), for more information.

### Tuning Parameters

#### 1. SQLNET.ORA timeout configuration

Tuning \$ORACLE\_HOME/network/admin/sqlnet.ora file on the Oracle Client

For proper function of the reactivate timer interval, one or more of the following parameters in sqlnet.ora file needs to be tuned:

- SQLNET.INBOUND\_CONNECT\_TIMEOUT
- SQLNET.OUTBOUND\_CONNECT\_TIMEOUT
- SQLNET.SEND\_TIMEOUT
- SQLNET.RECV\_TIMEOUT

Ensure that the ReactivateTimerInterval of ODBC/ODBC-Accounting remoteservers is greater than the timeout values configured in sqlnet.ora.

#### 2. AdditionalNativeOracleErrors connection lost error configuration

Whenever OCI remote server oracle connection encounters configured ORA error, Prime Access Registrar will disconnect the remote server and reactivate it after the ReactivateTimerInterval

#### Example

```
set /Radius/Advanced/AdditionalNativeOracleErrors
"31113,31114,12543,25408,25402,600,12502,12170,3135, 12518, 12526, 12528, 1089, 12547,
1041, 1092, 12537, 12514"
```

## SQL Definitions

SQLDefinitions lists the UserPasswordAttribute and one or more SQL statements, listed numerically in the order to be run. The UserPasswordAttribute represents a column in the database that contains users' password information. Individual SQLStatements are numbered SQL1 through SQL $n$  under SQLStatements, as shown in the following example:

```
SQLDefinition/
  UserPasswordAttribute = asdfjkl
  SQLStatements/
    SQL1/
    SQL2/
    SQL3/
    ...
```

The following example is an SQL statement used for Authentication and Authorization:

```
SQLStatements/
  SQL1
```

Name = SQL1  
 Type = **query** (mandatory, no default; must be query/procedure)  
 SQL = **SQL statement** (mandatory, no default)  
 ExecutionSequenceNumber = Sequence number for SQLStatement execution. (mandatory, no default and must be greater than zero).  
 MarkerList = UserName/SQL\_DATA\_TYPE ..... (mandatory, UserName must be defined)

For more information on stored procedures and stored functions, refer to [Oracle Stored Procedures](#), page 13-14.

Table 13-3 describes the SQL Statement parameters.

**Table 13-3 SQL Statement Parameters**

| Parameter               | Description   |
|-------------------------|---|
| Name                    | Name/number of SQL statement  |
| Type                    | Query (mandatory, no default value)   |
| SQL                     | SQL query statement   |
| ExecutionSequenceNumber | Sequence number for SQLStatement execution, must be greater than zero (mandatory, no default) |
| MarkerList              | Defines all markers for the query. MarkerList uses the format <i>UserName/SQL_DATA_TYPE</i> . |

## SQL Syntax Restrictions

You must observe the following SQL syntax restrictions in SQL queries for Prime Access Registrar.

1. The SQL statement must be in the format of SELECT ... FROM ... WHERE ..." (Statements might be in lowercase.)



**Note** 'WHERE' is compulsory in the SQL statement.

2. Stored procedures with return value must be in the "*begin ? := <Stored\_procedure\_name> (<IN/OUT Parameters>); end;*" format.
3. Stored procedures without return value can be in the "CALL <Stored\_procedure\_name> (<IN/OUT Parameters>)" format.
4. Any arguments to Oracle functions like *distinct*, *count* must be given within braces, as shown in the following example:

```
select distinct(attribute),password from profiles where username=?
```

The resulted column from *distinct(attribute)* will be put into *attribute* which can be used for ODBC Mappings. The actual result set from Oracle for this column would be named *distinct(attribute)*.

5. The column list in the SQL statement must be delimited with a comma (,) and any extra spaces between statements are ignored. Aliasing for column names in SQL is not allowed. SQLDefinition properties define the SQL you want to execute, as shown in the following example.

## Specifying More Than One Search Key

You can specify more than one search key for a table in the SQL SELECT. To do so, add another search criteria to the SQL statement and add the environment variable name to the MarkerList. For example, the following query and MarkerList can be used to look up a username and CLID match.

```
select password from user_table where username = ? and clid = ?
```

In this case, the marker list would look like this:

```
UserName/SQL_CHAR clid/SQL_CHAR
```

To configure the multiple entries in the MarkerList list, surround the entire string in double quotes like the following:

```
set MarkerList "UserName/SQL_CHAR CLID/SQL_CHAR"
```

To make this work, a variable called CLID must be in the environment dictionary. You can use a script to copy the appropriate value into the variable.

## ODBCToRadiusMappings/OCIToRadiusMappings

You configure ODBCToRadiusMappings or OCIToRadiusMappings with a list of *name/value* pairs where name is the name of the data store attribute to retrieve from the user record and the value is the name of the RADIUS attribute to set to the value of the data store attribute retrieved.

For example, use the following **aregcmd** command to set a value for the variable *Framed-IP-Address*:

```
set FramedIPAddress Framed-IP-Address
```

When the ODBCToRadiusMappings or OCIToRadiusMappings has this entry, the RemoteServer retrieves the attribute from the data store user entry for the specified user, uses the value returned, and sets the response variable *Framed-IP-Address* to that value.

When an SQL select statement returns more than one row for a column mapped under ODBCToRadiusMappings or OCIToRadiusMappings, multiple Radius attributes are created.

For example, consider the following SQL *select* statement with ciscoavpair configured to Cisco-AVPair under ODBCToRadiusMappings. The table.column syntax requires an SQL alias for the mapping to work, as shown in the following example:

```
SQLStatements/
SQL1/
select table1.abc as t1abc, password from table2 where username = ?
Mapping: t1abc = my_mapping
```

If two rows are returned for ciscoavpair column, two Cisco-AVPair attributes will be created.

## ODBCToEnvironmentMappings/OCIToEnvironmentMappings

Under ODBCToEnvironmentMappings or OCIToEnvironmentMappings there is a list of name and value pairs in which the name is the name of the data store attribute to retrieve from the user record, and the value is the name of the Environment variable to set to the value of the ODBC or OCI attribute retrieved.

For example, when the ODBCToEnvironmentMappings has the entry: group =User-Group, the RemoteServer retrieves the attribute from the ODBC user entry for the specified user, uses the value returned, and sets the environment variable User-Group to that value. When an SQL select statement returns more than one row for a column mapped under ODBCToEnvironmentMappings, the value for all rows is concatenated and assigned to the environment variable.

## ODBCToCheckItemMappings/OCItoCheckItemMappings

A list of ODBC or OCI *attribute/value* pairs which must be present in the RADIUS access request and must match, both name and value, for the check to pass.

For example, when the **ODBCToCheckItemMappings** or **OCItoCheckItemMappings** has the entry: **group = User-Group**, the Access Request must contain the attribute **group**, and it must be set to **User-Group**.

## Configuring an ODBC DataSource

ODBCDataSource is the name of the datasource to be used by the remote server. You configure ODBCDataSources under **/Radius/Advanced/ODBCDataSources**. Multiple remote servers can use the same ODBCDataSource.

Under the ODBCDataSource object definition, for ODBC a list defines **ODBC.ini** filename/value pairs for a connection. The list includes a Type field and a Driver field, different for each Driver and Data Source, to indicate its Driver and Data Source. Prime Access Registrar currently supports only the Easysoft Open Source Oracle Driver.

For OCI services, ODBCDataSource type should be 'oracle\_oci'. The following is an example configuration of ODBCDataSource for OCI services.

```
[ //localhost/Radius/Advanced/ODBCDataSources/gordon ]
Name = gordon
Description =
Type = oracle_oci
UserID = scott
Password = <encrypted>
DataBase = orcl.cisco.com
```

Table 13-4 describes the OCILib Open Source Oracle Driver options for OCI.

**Table 13-4** OCILib Open Source Oracle Driver Options for OCI

| Parameter | Description  |
|-----------|--|
| Name      | Name of the ODBCDataSource   |
| Type      | Required; must be Oracle_oci   |
| Database  | Required; Oracle Client configuration database name (no default value) |
| UserID    | Required; database username (no default value)                         |
| Password  | Optional user password; shown encrypted                                |

## Setting ODBC/OCI As Authentication and Authorization Service

Use **aregcmd** to configure the ODBC Service as the default authentication and authorization service under **//localhost/Radius** as in the following:

```
set DefaultAuthenticationService odbc-service
```

```
set DefaultAuthorizationService odbc-service
```

Use **aregcmd** to configure the OCI Service as the default authentication and authorization service under **//localhost/Radius** as in the following:

```
set DefaultAuthenticationService oci-service
```

```
set DefaultAuthorizationService oci-service
```

**Note**

---

When you use an ODBC or OCI service, configure the BackingStoreDiscThreshold property under **/Radius/Advanced** to ensure that the data generated by log files do not exceed the size limit configured.

---

## Setting ODBC/OCI As Accounting Service

Use **aregcmd** to configure the ODBC Service as the default accounting service under **//localhost /Radius** as in the following:

```
set DefaultAccountingService odbc-service
```

Use **aregcmd** to configure the OCI Service as the default authentication and authorization service under **//localhost /Radius** as in the following:

```
set set DefaultAccountingService oci-service
```

## Saving Your Configuration

When you use **aregcmd** to **save** your configuration, Prime Access Registrar attempts to validate the configuration, checks for all required parameters, and ensures there is no logic error. If the validation is successful, the configuration is saved to the MCD database. When you **reload**, Prime Access Registrar shuts down any current ODBC/OCI connections and builds new connections for the configured ODBC Data Sources.

## Oracle Stored Procedures

A stored procedure is a database procedure similar to other programming language procedures, which is contained within the database itself. A SQL Server stored procedure that contains one or more IN parameters are used to pass data into the stored procedure. Similarly, one or more OUT parameters in the stored procedure are used to return data back to the calling application. Prime Access Registrar supports Oracle stored procedures/functions with IN and OUT parameters only over the OCI interface.

For Authentication and Authorization, Prime Access Registrar supports both Stored Procedures and Stored Functions with the In/Out parameters and return value. In the configuration for the AA remote server, the UserPasswordAttribute value must be in the marker list for procedures.

For Accounting, Prime Access Registrar supports both Stored Procedures and Stored Functions with only the In parameters, and does not support return value and Out parameters.

The following are the examples for stored functions and procedures calling inside Prime Access Registrar:

```
Example format for stored functions with return value
SQL = "begin ? := stress (?);end;"
```

```
Example for stored procedures
SQL = " CALL Accounting_Request( ?,?,?)"
```

**Note**

Prime Access Registrar does not support, return value with the "call" format for the stored procedures.

The following shows an example configuration for OCI AA remote server:

```
[ //localhost/Radius/RemoteServers ]
Entries 1 to 2 from 2 total entries
Current filter: <all>

AA_OCI/

    Name = AA_OCI
    Description =
    Protocol = oci
    ReactivateTimerInterval = 60000
    Timeout = 1
    OCITimeOutCount = 2
    OCIActiveConnectionThresholdCount = 2
    OCIConnectionReactivationInterval = 30000
    DataSourceConnections = 4
    ODBCDataSource = VM011DB
    SNMPTrapIP =
    SNMPTrapPort = 1521
    KeepAliveTimerInterval = 3000
    SQLDefinition/
        UserPasswordAttribute = password
    SQLStatements/
        Entries 1 to 1 from 1 total entries
        Current filter: <all>

    sql1/
        Name = sql1
        Description =
        Type = procedure
        SQL = "call Access_Request(?,?)"
        ExecutionSequenceNumber = 1
        MarkerList = "UserName/SQL_CHAR password/SQL_OUT"
    OCIToRadiusMappings/
    OCIToEnvironmentMappings/
    OCIToCheckItemMappings/
```

The following shows an example configuration for OCI AAA remote server:

```
accounting/

    Name = accounting
    Description =
    Protocol = oci-accounting
    ReactivateTimerInterval = 6000
    Timeout = 3
    OCITimeOutCount = 2
    OCIActiveConnectionThresholdCount = 2
    OCIConnectionReactivationInterval = 3000
    DataSourceConnections = 4
    ODBCDataSource = VM011DB
    SNMPTrapIP =
    SNMPTrapPort = 1521
    KeepAliveTimerInterval = 0
    BufferAccountingPackets = FALSE
    MaximumBufferSize = "10 Megabytes"
```

```

NumberOfRetriesForBufferedPacket = 3
BackingStoreEnvironmentVariables =
UseLocalTimeZone = FALSE
OCIAutoCommit = TRUE
OCITransactionCount = 0
AttributeList =
Delimiter =
SQLDefinition/
  SQLStatements/
    Entries 1 to 1 from 1 total entries
    Current filter: <all>

sql/
  Name = sql
  Description =
  Type = procedure
  SQL = "call Accounting_Request(?,?)"
  ExecutionSequenceNumber = 1
  MarkerList = "UserName/SQL_CHAR Acct-Status-Type/SQL_CHAR"

```

**Note**


---

Prime Access Registrar supports Oracle stored procedures for OCI AA and OCI AAA remote servers.

---

## MySQL Support

Prime Access Registrar provides support for MySQL to query user records from a MySQL database and enables you to write accounting records into MySQL when using Oracle accounting. Prime Access Registrar has been tested with MySQL 5.0.90 and MyODBC 3.51.27 (reentrant).

This section contains the following topics:

- [MySQL Driver](#)
- [Configuring a MySQL Datasource](#)
- [Example Configuration](#)

## MySQL Driver

You can download the MySQL driver from the MySQL website at <http://mysql.com>. You can go directly to the driver download page using the following URL:

<http://dev.mysql.com/downloads/connector/odbc/3.51.html>

Save the downloaded file to a temporary location such as **/tmp**. Use commands to unzip and install the driver.

For better performance with mysql, add the following code to the odbcinst.ini file under the **/cisco-ar/odbc/etc** directory:

```

[mysql]
DRIVER=/usr/lib/libmyodbc3_r.so
Threading      = 1

```

libmyodbc3\_r.so is the driver location

## Configuring a MySQL Datasource

You require the following to configure a MySQL Datasource:

- ODBCDataSource object
- RemoteServer object
- ODBC service
- Default AA services

### Configuring a MySQL datasource

To configure the Prime Access Registrar server to query records form a MySQL database:

---

**Step 1** Log into the Prime Access Registrar server and launch **aregcmd**.

Log in as a user with administrative rights such as user **admin**.

**Step 2** Change directory to the **/Radius/Advanced/ODBCDataSources** and add a new ODBCDataSource.

```
cd /Radius/Advanced/ODBCDataSources

add mysql
```

**Step 3** Set the new ODBCDataSource type to myodbc.

```
cd mysql

[ //localhost/Radius/Advanced/ODBCDataSources/mysql ]
  Name = mysql
  Description =
  Type =
```

```
set type myodbc
```

The following is the default configuration for an ODBCDataSource object of type myodbc:

```
[ //localhost/Radius/Advanced/ODBCDataSources/mysql ]
  Name = mysql
  Description =
  Type = myodbc
  Driver =
  UserID =
  Password =
  DataBase =
  Server =
  Port = 3306
```

**Step 4** Set the Driver property to the path of the MyODBC library. Use a command like the following:

```
set driver /scratch/myodbc/libmyodbc3_r.so
```

**Step 5** Set the UserID property to a valid username for the MyODBC database and provide a valid password for this user.

```
set userid ar-mysql-user

set password biscuit
```

- Step 6** Provide a DataBase name and the name of the Prime Access Registrar RemoteServer object to associate with the ODBCDataSource.
- ```
set database database_name

set server remote_server_name
```
- Step 7** Change directory to **/Radius/RemoteServers** and add a RemoteServer object to associate with the new ODBCDataSource.
- ```
cd /Radius/RemoteServers

add mysql
```
- Step 8** Change directory to the new RemoteServer and set its protocol to `odbc`.
- ```
cd mysql

set protocol odbc
```
- Step 9** Set the ODBCDataSource property to the name of the ODBCDataSource to associate with this RemoteServer object.
- ```
set ODBCDataSource mysql
```
- Step 10** Change directory to **/Radius/Services** and add an ODBC service as described in [Configuring an ODBC/OCI Service, page 13-6](#).
- Step 11** Change directory to **/Radius** and set the DefaultAuthenticationService and DefaultAuthorizationService properties to the ODBC service added in the previous step.
- 

## Example Configuration

The following shows an example configuration for a MySQL ODBC data source. See [Configuring an ODBC DataSource, page 13-13](#) for more information.

```
[ //localhost/Radius/Advanced/ODBCDataSources/mysql ]
  Name = mysql
  Type = myodbc
  Driver = /tmp/libmyodbc3_r.so
  UserID = mysql
  Password = <encrypted>
  DataBase = test
  Server = mysql-a
  Port = 3306
```

The following shows an example configuration for a RemoteServer. See [Configuring an ODBC/OCI RemoteServer, page 13-7](#) for more information.

```
[ //localhost/Radius/RemoteServers/mysql-a ]
  Name = mysql
  Description =
  Protocol = odbc
  ReactivateTimerInterval = 300000
  Timeout = 15
  DataSourceConnections = 8
  ODBCDataSource = mysql
```

```

KeepAliveTimerInterval = 0
SQLDefinition/
UserPasswordAttribute = asdfjkl
SQLStatements/
  SQL1/
    Name = SQL1
    Type = query (mandatory, no default; must be query)
    SQL = SQL statement (mandatory, no default)
    ExecutionSequenceNumber = Sequence number for SQLStatement
    execution.(mandatory, no default and must be greater than zero).
    MarkerList = UserName/SQL_DATA_TYPE ..... (mandatory, UserName must be defined)
  SQL2/
  SQL3/
ODBCToRadiusMappings/
ODBCToEnvironmentMappings/
ODBCToCheckItemMappings/

```

The following shows an example configuration for an ODBC service. See [Configuring an ODBC/OCI Service, page 13-6](#) for more information.

```

[ //localhost/Radius/Services/ODBC ]
  Name = ODBC
  Description =
  Type = ODBC
  IncomingScript~ =
  OutgoingScript~ =
  OutagePolicy~ = RejectAll
  OutageScript~ =
  MultipleServersPolicy = Failover
  RemoteServers/
    1. mysql-a

```

The following shows an example configuration where the DefaultAuthenticationService and DefaultAuthorizationService properties have been set to the ODBC service.

```

[ //localhost/Radius ]
  Name = Radius
  Description =
  Version = 7.2.0.0
  IncomingScript~ =
  OutgoingScript~ =
  DefaultAuthenticationService~ = ODBC
  DefaultAuthorizationService~ = ODBC

```





## SIGTRAN-M3UA

SIGTRAN, a working group of the Internet Engineering Task Force (IETF), has defined a protocol for the transport of real-time signaling data over IP networks. Cisco Prime Access Registrar (Prime Access Registrar) supports SS7 messaging over IP (SS7oIP) via SIGTRAN-M3UA, a new transport layer which leverages Stream Control Transmission Protocol (SCTP). Prime Access Registrar supports SIGTRAN-M3UA to fetch the authentication vectors from HLR, which is required for EAP-AKA/EAP-SIM authentication.

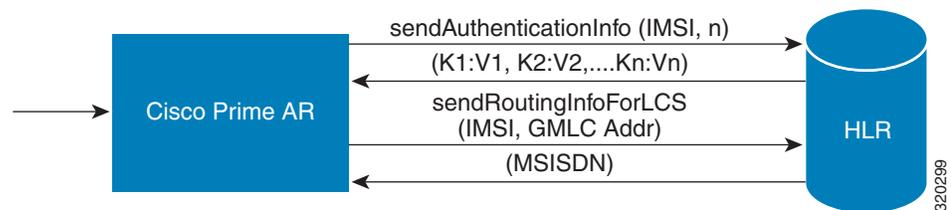


**Note**

You have SIGTRAN-M3UA interface support in addition to the existing SUA interface support.

The EAP-AKA and EAP-SIM authentication service is extended to use M3UA. When using M3UA service for authentication, the subscriber identity (IMSI) is used to send a request to HLR and receives information from HLR containing the authentication information for authenticating an user. The authentication service initiates a request to the SIGTRAN server using IMSI, which retrieves the configured number of authentication vectors from HLR, i.e Triplets or Quintets.

**Figure 14-1**     **MAP Service**



The Prime Access Registrar server initiates the MAP service. After enabling the MAP service, the Prime Access Registrar server sends a `sendAuthenticationInfo` request that contains IMSI and the number of requested authentication vectors to HLR. The HLR sends a response containing the requested vectors information to Prime Access Registrar. Next, the Prime Access Registrar server sends a `sendRoutingInfoForLCS` request that contains IMSI and the GMLC address to HLR. The HLR sends a response containing the MSISDN information for authenticating the mobile subscribers.

Prime Access Registrar provides `map-restore-data` authentication support for `m3ua` services.

Prime Access Registrar supports multiple remote servers with the protocol type, SIGTRAN-M3UA. However, Prime Access Registrar validates and ensures the following when multiple remote servers are available:

- The source port is different for all the remote servers.
- If Origin Point Code (OPC) is different, the routing context is also different for all the remote servers.
- The Destination Point Code (DPC) is different for all the remote servers.
- The NetworkVariant, SubServiceField (SSF), TCAPVariant, NetworkAppearance, and NetworkIndicator values are the same for all the remote servers.

This section describes the following:

- [Prerequisites to SIGTRAN-M3UA](#)
- [Configuring EAP-AKA/EAP-SIM with SIGTRAN-M3UA](#)
- [Configuring M3UA Service](#)
- [Blacklisting Support for SIGTRAN-M3UA Remote Server, page 14-21](#)
- [Support for SCTP Multihoming in SIGTRAN-M3UA, page 14-21](#)
- [Tuning Global SIGTRAN Parameters, page 14-22](#)
- [SIGTRAN-M3UA Logs, page 14-24](#)

## Prerequisites to SIGTRAN-M3UA

Before enabling the SIGTRAN-M3UA remote server, you must do the following:

- ensure that LKSCTP is not available in the Prime Access Registrar server.
- ensure to restart the Prime Access Registrar server whenever you make any configuration changes.
- ensure that you have the 32-bit rpm files for the relevant RHEL OS versions while installing the Cisco Prime Access Registrar. For the list of required rpms for the relevant OS versions, see [Required 32-bit rpms for Relevant RHEL OS Versions](#).




---

**Note** You must install the rpm versions relevant to the RHEL OS versions while installing the Prime Access Registrar.

---

- ensure that the ‘bc’ command (which is an arbitrary precision calculator language) is present while installing Prime Access Registrar in a Linux machine. If the ‘bc’ command is not present, install the relevant rpm such as bc-1.06.95-1.el6.x86\_64 on that machine.
- ensure that you have the following packages while installing the Prime Access Registrar:
  - gcc version-3.4.6
  - gdome-config-0.8.1




---

**Note** You need to build the gdome-config-0.8.1 package to make it available. For more information, see [Building gdome Package, page 14-3](#)

---

- xml2-config-2.6.23

- pkg-config-0.15.0
- glib-2.30
- gtk-2.41
- libxml-2.2.6.20

#### Required 32-bit rpms for Relevant RHEL OS Versions

| rpm                | RHEL OS Version 6.4 | RHEL OS Version 6.6 | RHEL OS Version 7.0 |
|--------------------|---------------------|---------------------|---------------------|
| glibc              | Yes                 | Yes                 | Yes                 |
| gdome2             | Yes                 | Yes                 | Yes                 |
| glib               | Yes                 | Yes                 | Yes                 |
| glib2              | Yes                 | Yes                 | Yes                 |
| libgcc             | Yes                 | Yes                 | Yes                 |
| libstdc++          | Yes                 | Yes                 | Yes                 |
| libxml2            | Yes                 | Yes                 | Yes                 |
| ncurses            | No                  | No                  | No                  |
| nspr               | Yes                 | Yes                 | Yes                 |
| nss                | No                  | No                  | No                  |
| zlib               | Yes                 | Yes                 | Yes                 |
| nss-softokn-freebl | Yes                 | Yes                 | Yes                 |
| ncurses-libs       | Yes                 | Yes                 | Yes                 |
| nss-util           | Yes                 | Yes                 | Yes                 |
| gamin              | Yes                 | Yes                 | Yes                 |
| libselinux         | Yes                 | Yes                 | Yes                 |

#### Building gdome Package

To build gdome-config-0.8.1 package:

**Step 1** Download gdome2-0.8.1.tar.gz package from the location <http://gdome2.cs.unibo.it/#downloads>.

**Step 2** Execute the following command:

```
gunzip gdome2-0.8.1.tar.gz
```

**Step 3** Untar the package using the following command:

```
tar -xvf gdome2-0.8.1.tar
```

**Step 4** Use the `cd` command to move into the package obtained from [Step 3](#).

**Step 5** Execute the following commands:

```
./configure --prefix=<GdomeInstallPath> --with-glib-prefix=<GlibInstalledDirectory>
make
make install
```

Where,

- `GdomeInstallPath` specifies where the Gdome libraries must be placed.
- `GlibInstalledDirectory` specifies which directory the Glib libraries reside in the filesystem.

**Step 6** Now `gdome` libraries will be available in the location `GdomeInstallPath`.

---

## Configuring EAP-AKA/EAP-SIM with SIGTRAN-M3UA

You can use `aregcmd` to create and configure the service of type `eap-aka` or `eap-sim`, see [EAP-AKA](#) or [EAP-SIM](#) for more information.

To configure EAP-AKA service with SIGTRAN-M3UA remote server:

---

**Step 1** Launch `aregcmd`.

**Step 2** Create an EAP-AKA service.

```
cd /Radius/Services
add eap-aka-service
```

**Step 3** Set type as `eap-aka`.

```
set eap-aka
```

**Step 4** Add `m3ua` remote server in the `remoteServers`

```
cd remoteServers/
Set 1 m3ua
```

---

The following shows an example configuration for EAP-AKA service with SIGTRAN-M3UA remote server support, see [Table 5-1](#) to know more about EAP-AKA service properties.

```
[ //localhost/Radius/Services ]
Entries 1 to 2 from 2 total entries
Current filter: <all>

eap-aka/
  Name = eap-aka
  Description =
  Type = eap-aka
  AlwaysRequestIdentity = False
  EnableIdentityPrivacy = False
  PseudonymSecret = <encrypted>
  PseudonymRenewtime = "24 Hours"
  PseudonymLifetime = Forever
  Generate3GPPCompliantPseudonym = False
  EnableReauthentication = False
  MaximumReauthentications = 16
  ReauthenticationTimeout = 3600
  ReauthenticationRealm =
  AuthenticationTimeout = 120
  QuintetGenerationScript~ =
```

```

UseProtectedResults = False
SendReAuthIDInAccept = False
Subscriber_DBLookup = SIGTRAN-M3UA
FetchAuthorizationInfo = FALSE
MultipleServersPolicy = Failover
IncomingScript~ =
OutgoingScript~ =
OutageScript~ =
RemoteServers/

```

To configure EAP-SIM service with SIGTRAN-M3UA remote server:

- 
- Step 1** Launch **aregcmd**.
- Step 2** Create an EAP-SIM service.
- ```

cd /Radius/Services

add eap-sim-service

```
- Step 3** Set type as eap-sim.
- ```

set eap-sim

```
- Step 4** Add m3ua remote server in the remoteServers
- ```

cd remoteServers

Set 1 m3ua

```
- 

The following shows an example configuration for EAP-SIM service with SIGTRAN-M3UA remote server support. See [Table 5-6](#) to know more about EAP-SIM service properties.

```

eap-sim/
Name = eap-sim
Description =
Type = eap-sim
NumberOfTriplets = 2
UseSimDemoTriplets = False
AlwaysRequestIdentity = False
EnableIdentityPrivacy = False
PseudonymSecret = <encrypted>
PseudonymRenewtime = "24 Hours"
PseudonymLifetime = Forever
Generate3GPPCompliantPseudonym = False
EnableReauthentication = False
MaximumReauthentications = 16
ReauthenticationTimeout = 3600
ReauthenticationRealm =
TripletCacheTimeout = 0
AuthenticationTimeout = 120
UseProtectedResults = False
SendReAuthIDInAccept = False
SubscriberDBLookup = SIGTRAN-M3UA
FetchAuthorizationInfo = FALSE
MultipleServersPolicy = Failover
IncomingScript~ =
OutgoingScript~ =
OutageScript~ =

```

```
RemoteServers/
```

**Note**

After enabling the SIGTRAN-M3UA remote server, you must ensure to restart the Prime Access Registrar server whenever you make any configuration changes.

**Note**

If you set `FetchAuthorizationInfo` as `TRUE` for EAP-AKA or EAP-SIM service for SIGTRAN-M3UA in Prime Access Registrar, it fetches the MSISDN information from HLR in response. The following is an example script for reading the MSISDN information from the response,

```
proc MapMSISDN {request response environ} {  
  $environ get AuthorizationInfo  
}
```

**Configuring SIGTRAN-M3UA Remote Server**

You can configure the SIGTRAN-M3UA remoteserver under **/Radius/RemoteServers**.

To configure the SIGTRAN-M3UA remote server:

**Step 1** Launch `aregcmd`.

**Step 2** Create sigtran-m3ua remote server.

```
cd /r/remoteservers/  
add M3UA  
cd M3UA  
set protocol sigtran-m3ua
```

**Step 3** Set the `Subscriber_DBLookup`.

```
set Subscriber_DBLookup SIGTRAN-M3UA
```

**Step 4** Set the port of the HLR.

```
set DestinationPort 2905
```

**Step 5** Set the port for the source.

```
set SourcePort 2905
```

**Step 6** Set the reactivate timer interval for the remote server.

```
Set the reactivatetimerinterval.
```

**Step 7** Set the subsystem number for the local.

```
set LocalSubSystemNumber 149
```




---

**Note** Prime Access Registrar supports the following local Sub System Numbers (SSNs) by default:  
 SGSN (149)  
 VLR (7)  
 GMLC (145)

---

**Step 8** Set routingindicator.

**Set routingindicator rte\_gt**

**Step 9** Set mlcnumber.

**Set mlcnumber**

**Step 10** Set routingparameters.

**cd routingparameters/**

**set OriginPointCode 2**

**set DestinationPointCode 4**

**set RemoteSubSystemNumber 6**

**set OPCMask 16383**

**set DPCMask 16383**

**set RoutingContext 11**

**Step 11** Set the source and destination gt parameters.

**Step 12** Set the numbering plan, encoding scheme, format, and digits for source.

**Step 13** Set the numbering plan, encoding scheme, format, and digits for destination.

---

## ANSI Support for SIGTRAN

Prime Access Registrar provides ANSI variant support apart from ITU variants in SIGTRAN stack for EAP-SIM and EAP-AKA services to M3UA.

While using this service for authentication, the subscriber identity (IMSI) is obtained from the request. Using this IMSI, the authentication service initiates a request to the SIGTRAN server. This request is to retrieve the configured number of authentication vectors (triplets/quintets) for the IMSI.

The remote SIGTRAN server initiates the IS41 service primitive Authentication Data request with IMSI and number of requested authentication vectors. This will retrieve the authentication vectors from HLR which will be used by the authentication service for authenticating the mobile subscriber.




---

**Note** Prime Access Registrar supports either ITU or ANSI variant in one running instance. Both the variants are not supported simultaneously.

---

The following shows an example configuration of SIGTRAN-M3UA remote server with ITU variant:

```
[ //localhost/Radius/RemoteServers/m3ua ]
  Name = m3ua
  Description =
  Protocol = sigtran-m3ua)
  SourcePort = 2905
  LocalSubSystemNumber = 149
  DestinationPort = 2905
  IMSITranslationScript~ =
  GlobalTitleTranslationScript~ = setGT
  Timeout = 15
  ReactivateTimerInterval = 2000
  LimitOutstandingRequests = FALSE
  MaxOutstandingRequests = 0
  MaxRetries = 3
  MAPVersion = 2
  NetworkVariant = ITU
  SubServiceField = NAT
  TCAPVariant = ITU96
  NetworkAppearance = 1
  NetworkIndicator = NAT
  MLCNumber = 123456789012345
  TrafficMode = LOADSHARE
  LoadShareMode = SLS
  RoutingIndicator = RTE_GT
  RoutingParameters/
    OriginPointCode = 2
    DestinationPointCode = 4
    RemoteSubSystemNumber = 6
    OPCMask = 16383
    DPCMask = 16383
    ServiceIndicatorOctet = 0
    RoutingContext = 11
  SourceGTAddress/
    SourceGTDigits = 919845071842
    SourceGTFormat = GTFRMT_4
    SourceNatureofAddress = INTNUM
    SourceTranslationType = 0
    SourceNumberingPlan = ISDN
    SourceEncodingScheme = BCDEVEN
  DestinationGTAddress/
    DestGTDigits = 919845071842
    DestGTFormat = GTFRMT_4
    DestNatureofAddress = INTNUM
    DestTranslationType = 0
    DestNumberingPlan = ISDN
    DestEncodingScheme = BCDEVEN
```

Table 14-1 describes SIGTRAN-M3UA remote server properties.

**Table 14-1**      **SIGTRAN-M3UA Stack Properties**

Property	Description
Name	Required; inherited from the upper directory.
Description	An optional description of the service.
Protocol	Represents the type of remote server. The value should be SIG-TRAN-M3UA.
SourcePort	The port number in which Prime Access Registrar is installed for M3UA transactions.

**Table 14-1 SIGTRAN-M3UA Stack Properties (continued)**

Property	Description
LocalSubSystemNumber	The local sub system number is set as 149 by default.
DestinationPort	The destination port number to which Prime Access Registrar connects.
IMSITranslationScript	The scripting point is used to modify the IMSI based on the requirement before sending the request to STP/HLR.
GlobalTitleTranslationScript	This is used to specify the name of the script which is responsible for translating IMSI to Global Title Address (GTA).  You can choose to configure blacklisting as part of the global title translation script for SIGTRAN-M3UA remote server. For more information about blacklisting, see .
Timeout	Specifies the time (in seconds) to wait before an authentication request times out; defaults to 15.
MaxTimeOuts	Maximum number of timeouts allowed for the remote server.
MaxSessionLimit	Maximum number of sessions allowed for the remote server.
ReactivateTimerInterval	Specifies the time interval (in milliseconds) to activate an inactive server; defaults to 300000 ms (which is 5 minutes).
LimitOutstandingRequests	Required; the default is FALSE. Prime Access Registrar uses this property in conjunction with the MaxOutstandingRequests property to tune the RADIUS server's use of the HLR.  When you set this property to TRUE, the number of outstanding requests for this RemoteServer is limited to the value you specified in MaxOutstandingRequests. When the number of requests exceeds this number, Prime Access Registrar queues the remaining requests, and sends them as soon as the number of outstanding requests drops to this number.
MaxOutstandingRequests	Required when you have set the LimitOutstandingRequests to TRUE. The number you specify, which must be greater than zero, determines the maximum number of outstanding requests allowed for this remote server.
TrafficMode	The mode of the traffic for the HLR. The possible values are LOADSHARE or ACTSTANDBY.
LoadShareMode	Required. The TrafficMode is set as LOADSHARE, which is a type of load sharing scheme.  When there is more than one associations with HLR, then the load sharing is set as Signaling Link Selection (SLS). SLS is done based on a simple round-robin basis.
MAPVersion	The version of the MAP. The possible values are 2 or 3. Specify the MAP version that the HLR supports, i.e, 2 or 3 during the configuration.
NetworkVariant	Required. Choose ITU or ANSI to represent the network variant switch.

Table 14-1 SIGTRAN-M3UA Stack Properties (continued)

Property	Description
SubServiceField	Specifies the type of network to which this SAP belongs. The possible options are: <ul style="list-style-type: none"> <li>• INT—represents international network</li> <li>• NAT—represents national network</li> <li>• RESERVE—represents reserved network</li> <li>• SPARE—represents spare network</li> </ul>
SCCPVariant	The Signaling Connection Control Part (SCCP) variant of the Global Title: <ul style="list-style-type: none"> <li>• Select <b>ITU88</b>, <b>ITU92</b>, or <b>ITU96</b>, if NetworkVariant is set to ITU.</li> <li>• Select <b>ANS88</b>, <b>ANS92</b>, or <b>ANS96</b>, if NetworkVariant is set to ANS.</li> </ul>
TCAPVariant	Required; represents the name of the tcap network variant switch. The possible options are ITU88, ITU92, or ITU96.
NetworkAppearance	Required. A parameter that represents network appearance in the M3UA packet. Value ranges from 0-2147483647 and the default value is 1. This is optional as per the RFC 4666 ( <a href="http://tools.ietf.org/html/rfc4666">http://tools.ietf.org/html/rfc4666</a> .) You can set this value to 0 to remove network appearance from the data packet.
NetworkIndicator	The network indicator used in SCCP address. The possible options are NAT and INT which represents international network and national network respectively.
MLCNumber	Required, if you select FetchAuthorizationInfo as True in EAP-AKA or EAP-SIM services. Also, required for M3UA service for fetching the MSISDN from the HLR. This is the map layer network node number by which the HLR identifies the Prime Access Registrar in the network. The MLC number is configured in E.164 format.   <b>Note</b> MLC is a max-15 digit number.
RoutingIndicator	Required; represents the routing indicator. The possible values are Route on Global Title (RTE_GT) or Route on Sub System Number (RTE_SSN). You can use either RTE_GT or RTE_SSN value to route the packets for HLR.
<b>RoutingParameters</b>	
OriginPointCode	Required; represents the originating point of a message in a signaling network. The value ranges from 0-16777215.  This value must be less than OPCMask.
DestinationPointCode	Required; represents the destination address of a signaling point in a SS7 network.  This value must be less than DPCMask.
RemoteSubSystemNumber	Required; represents the sub system number of the remote server. The RemoteSubSystemNumber is set as 6 by default.

**Table 14-1 SIGTRAN-M3UA Stack Properties (continued)**

Property	Description
OPCMask	Represents the wild card mask for the origin point code. The value ranges from 0-16777215. Default value is 16383 for ITU and16777215 for ANSI.
DPCMask	Represents the wild card mask for the destination point code. The value ranges from 0-16777215. Default value is 16383 for ITU and16777215 for ANSI.
ServiceIndicatorOctet	Represents the service identifier octet. The value ranges from 0-255.
RoutingContext	Required; represents the routing context which ranges from 0-16777215.
<b>SourceIPAddresses</b>	
add 1, add 2,...	Represent the multiple source IP addresses configured on the remote server.
<b>DestinationIPAddresses</b>	
add 1, add 2,...	Represent the multiple destination IP addresses configured on the remote server.
<b>SourceGTAddress</b>	
The following fields are displayed only when you set RTE_GT as RoutingIndicator.	
SourceGTDigits	Required; an unique number to identify the source.
SourceGTFormat	Required; represents the format of the global translation (GT) rule. The possible values are GTFRMT_0, GTFRMT_1, GTFRMT_2, GTFRMT_3, GTFRMT_4, or GTFRMT_5. The GT format is GTFRMT_0, GTFRMT_1, or GTFRMT_2 for ANSI variant. GTFRMT_0 is the default format for both ANSI and ITU variants.
SourceNatureofAddress	Required; represents the type of the source address. The possible values are ADDR_NOTPRSNT (Address not present), SUBNUM (Subscriber number), NATSIGNUM (National significant number), or INTNUM (International number.)
SourceTranslationType	Required; represents the type of translation. The possible values ranges from 0-255.
SourceNumberingPlan	Required; represents the numbering plan of the network that the subscriber uses. For example, land mobile numbering plan, ISDN mobile numbering plan, private or network specific numbering plan.
SourceEncodingScheme	Required; represents the BCD encoding scheme. The possible values are UNKN (Unknown), BCDODD (BCD Odd), BCDEVEN (BCD Even), or NWSPEC (National specific). This must be set based on the length of the GT.
<b>DestinationGTAddress</b>	
The following fields are displayed only when you set RTE_GT as RoutingIndicator.	
DestGTDigits	Required; an unique number to identify the destination.

Table 14-1 SIGTRAN-M3UA Stack Properties (continued)

Property	Description
DestGTFormat	Required; represents the format of the global translation (GT) rule. The possible values are GTFRMT_0, GTFRMT_1, GTFRMT_2, GTFRMT_3, GTFRMT_4, or GTFRMT_5.  The GT format is GTFRMT_0, GTFRMT_1, or GTFRMT_2 for ANSI variant. GTFRMT_0 is the default format for both ANSI and ITU variants.
DestNatureofAddress	Required; represents the type of the destination address. The possible values are ADDR_NOTPRSNT (Address not present), SUBNUM (Subscriber number), NATSIGNUM (National significant number), or INTNUM (International number.)
DestTranslationType	Required; represents the type of translation. The possible values ranges from 0-255.
DestNumberingPlan	Required; represents the numbering plan of the network that the subscriber uses. For example, Land mobile numbering plan, ISDN mobile numbering plan, private or network specific numbering plan. Possible values are DATA, GENERIC, ISDN, ISDNMOB, LANMOB, MARMOB, NWSPEC, TEL, TELEX, and UNKN.
DestEncodingScheme	Required; represents the BCD encoding scheme. The possible values are UNKN (Unknown), BCDODD (BCD Odd), BCDEVEN (BCD Even), or NWSPEC (National specific). This must be set based on the length of the GT.

The following shows an example configuration of SIGTRAN-M3UA remote server with ANSI variant:

```
[ //localhost/Radius/RemoteServers ]
  Entries 1 to 1 from 1 total entries
  Current filter: <all>

  STP/
    Name =STP
    Description =
    Protocol = Sigtran-m3ua
    SourcePort = 2905
    LocalSubSystemNumber = 149
    DestinationPort = 2905
    IMSITranslationScript~ =
    Timeout = 15
    MaxTimeOuts = 200
    MaxSessionLimit = 0
    ReactivateTimerInterval = 2000
    LimitOutstandingRequests = FALSE
    MaxOutstandingRequests = 0
    MAPVersion = 2
    NetworkVariant = ANS
    SubServiceField = NAT
    SCCPVariant = ANS92
    TCAPVariant = ITU96
    NetworkAppearance = 1
    NetworkIndicator = NAT
    MLCNumber = 123456789012345
    TrafficMode = LOADSHARE
    LoadShareMode = SLS
    RoutingIndicator = RTE_GT
    GlobalTitleTranslationScript~ =
```

```

MaskPointCode = FALSE
RoutingParameters/
  OriginPointCode = 13967019
  DestinationPointCode = 13966849
  RemoteSubSystemNumber = 6
  OPCMask = 16777215
  DPCMask = 16777215
  ServiceIndicatorOctet = 3
  RoutingContext = 11
SourceIPAddresses/
  1. 10.81.78.142
DestinationIPAddresses/
  1. 10.81.78.145
SourceGTAddress/
  SourceGTDigits = 919845071842
  SourceGTFormat = GTFRMT_2
  SourceTranslationType = 10
DestinationGTAddress/
  DestGTDigits = 919845071842
  AdditionalDestGTDigits = 9198,2011
  DestGTFormat = GTFRMT_2
  DestTranslationType = 9

```

## Configuring M3UA Service

Prime Access Registrar supports the M3UA service, which is used to fetch MSISDN from IMSI or vice versa through RADIUS packets.

To configure the M3UA service with SIGTRAN-M3UA remote server:

---

**Step 1** Launch **aregcmd**.

**Step 2** Create an M3UA service.

```

cd /Radius/Services

add FetchAuthInfo

```

**Step 3** Set the type as M3UA.

```

set type M3UA

```

**Step 4** Set **AuthorizationInfoLookUp** to one of the following:

- MSISDN-IMSI—To fetch MSISDN in the request and send IMSI in the response to the HLR.
- IMSI-MSISDN—To fetch IMSI in the request and send MSISDN in the response to the HLR.




---

**Note** See [Example Configuration, page 14-14](#) for a sample configuration with

---

- Map-Restore—To fetch the profile information of a subscriber from the HLR. For more information on configuring the M3UA service with Map Restore Data authorization, see [Configuring M3UA Service with Map Restore Data Authorization, page 14-14](#).

```

set AuthorizationInfoLookUp IMSI-MSISDN

```

**Step 5** Add M3UA remote server in the remoteServers.

```
cd remoteServers
```

```
Set 1 m3ua
```

#### Example Configuration

The following shows an example configuration of the M3UA service:

```
[ //localhost/Radius/Services/test ]
  Name = test
  Description =
  Type = m3ua
  IncomingScript~ =
  OutgoingScript~ =
  OutageScript~ =
  OutagePolicy~ = RejectAll
  AuthorizationInfoLookUp = IMSI-MSISDN
  RemoteServers/
```

## Configuring M3UA Service with Map Restore Data Authorization

Prime Access Registrar provides the Map Restore Data functionality to fetch the profile information of a subscriber from the HLR.

This topic contains the following sections:

- [Map Restore Data Authorization Flow, page 14-14](#)
- [CS Insert Subscriber Data Structure, page 14-15](#)
- [CLI Configuration for Map-Restore-Data, page 14-16](#)

### Map Restore Data Authorization Flow

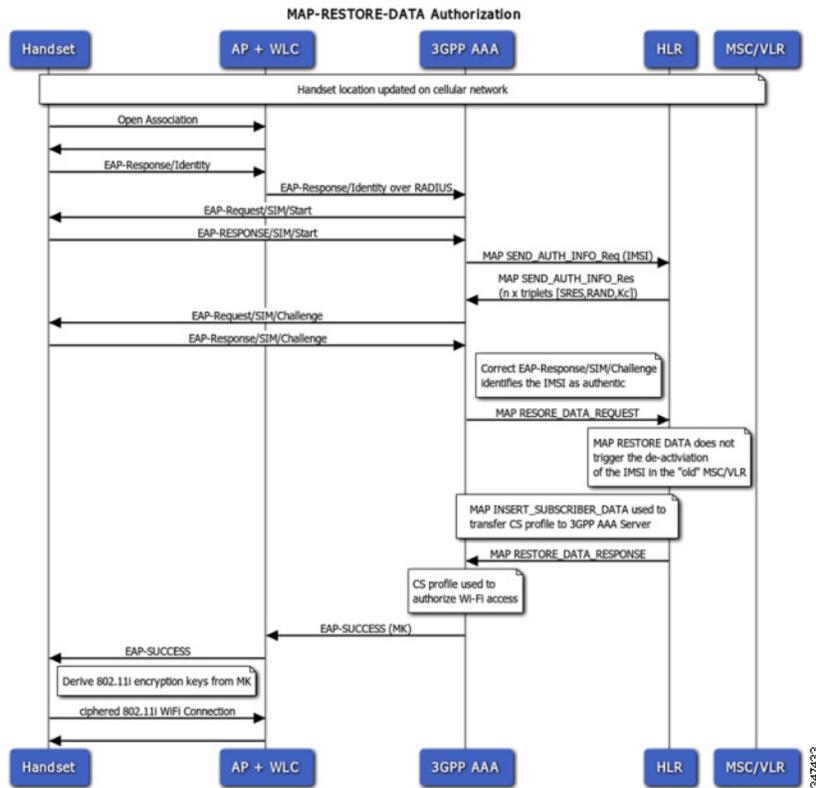
Prime Access Registrar sends a MAP\_SEND\_AUTH\_INFO request to HLR on receiving EAP-SIM / EAP-AKA authentication request and fetches the authentication vectors in MAP\_SEND\_AUTH\_INFO\_RES message. Prime Access Registrar checks the IMSI and if it is authentic, sends a MAP\_RESTORE\_DATA\_REQUEST to fetch the profile information from the HLR. HLR then responds with MAP\_INSERT\_SUBSCRIBER\_DATA request to Prime Access Registrar. The request contains the circuit switched (CS) profile information for a subscriber.

Prime Access Registrar server stores the profile information based on the ProfileInfo configuration and sends a MAP\_INSERT\_SUBSCRIBER\_DATA\_RESPONSE to HLR. HLR responds with MAP\_RESTORE\_DATA\_RESPONSE to Prime Access Registrar. After successful acknowledgment of MAP\_RESTORE\_DATA, Prime Access Registrar server maps the fetched profile through RestoreDataMappings to any of the environment variables configured by the user. The CS profile used to authorize WI-FI access which is fetched from HLR can be transported to access point in any of the radius attribute.

The mapping of the values in the response to a profile is possible based on the configuration in the profilemappings configuration.

[Figure 14-2](#) represents the Map-Restore-Data message flow between Prime Access Registrar and HLR.

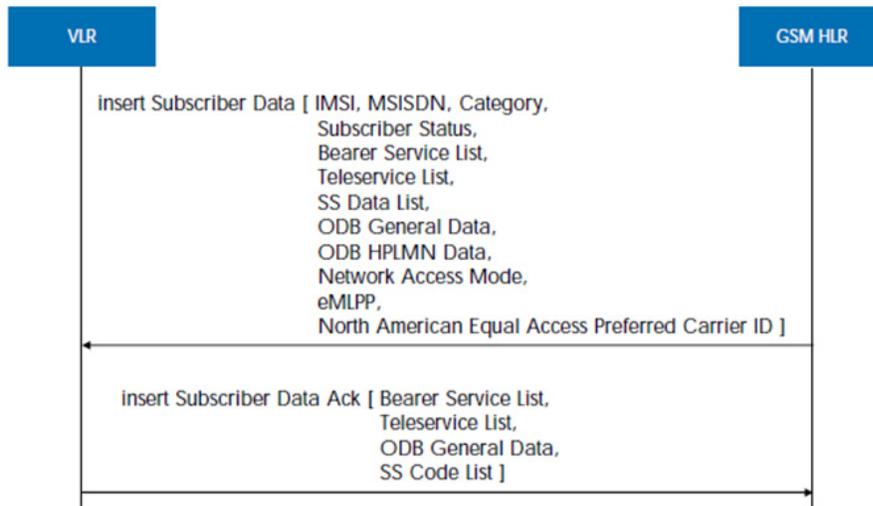
Figure 14-2 Map-Restore-Data Authorization Flow



### CS Insert Subscriber Data Structure

Figure 14-3 shows the parameters fetched by Prime Access Registrar on receipt of the subscriber data request.

Figure 14-3 CS Insert Subscriber Data Structure



## CLI Configuration for Map-Restore-Data

If you set `AuthorizationInfoLookUp` to **Map-Restore**, two additional properties `ProfileMappings` and `RestoreDataMappings` are displayed.

The restore data mapping parameters include LSA information, LCS information, and subscriber data. You can configure an index with a value or a range to fetch one or more properties from the subscriber data.

The following is an example configuration of an M3UA service with Map-Restore-Data authorization:

```
[ //localhost/Radius/Services/serv1 ]
  Name = serv1
  Description =
  Type = m3ua
  IncomingScript~ =
  OutgoingScript~ =
  OutageScript~ =
  OutagePolicy~ = RejectAll
  AuthorizationInfoLookUp = MAP-RESTORE
  RemoteServers/
    1. server1
  RestoreDataMappings/
    IMSI = imsi
    Naea-PreferredCI = naea
    RoamingRestrictedInSgsnDueToUnsupportedFeature =
    NetworkAccessMode =
    LMUIndicator =
    ISTAlertTimer =
    SuperChargerSupportedInHLR =
    CSAllocationRetentionPriority =
    ChargingCharacteristics =
    AccessRestrictionData =
    UE-ReachabilityRequestIndicator =
    Category =
  LSAInformation/
    CompleteDataListIncluded = completedatalist
    LSAOnlyAccessIndicator =
  LSADataList/
    Index = 6
    LSAIdentity = lsaid
    LSAAttributes = lsaattrib
    LSAActiveModeIndicator = activemode
  SubscriberData/
    MSISDN = msisdn
    SubscriberStatus = substatus
    RoamingRestrictionDueToUnsupportedFeature =
  BearerServiceList/
    Index = 6-10
    BearerService = bearsrvc
  TeleServiceList/
    Index =
    TeleService =
  ProvisionedSS/
    Index = 4-6
  ForwardingInfo/
    FI-SS-Code = fsscode
  ForFeatureList/
    Index = 7-10
    FF-SS-Status = ffsstatus
    ForwardedToNumber =
    ForwardedToSubaddress =
    ForwardingOptions =
```

```

        NoReplyConditionTime =
        LongForwardedToNumber =
        BasicService/
            BS-Ext-BearerService = bsextbsservice
            BS-Ext-Teleservice = bsextteleservice
    CallBarringInfo/
        CB-SS-Code =
        CallBarFeatureList/
            Index =
            CB-SS-Status =
            BasicService/
                CB-Ext-BearerService =
                CB-Ext-Teleservice =
        CugInfo/
            CugSubList/
                Index =
                CugSubscription/
                    Cug-Index =
                    cug-Interlock =
                    IntraCUG-Options =
                    BasicServiceGroupList/
                        Index =
                        CUG-Ext-BearerService =
                        CUG-Ext-Teleservice =
        CugInformation/
            Cug-FeatureList/
                Index =
                CUG-Feature/
                    BasicService.Ext-BearerService =
                    PreferentialCUG-Indicator =
                    InterCUG-Restrictions =
    SS-Data/
        SSD-SS-Code =
        SSD-SS-Status =
        SS-SubscriptionOption/
            CliRestrictionOption =
            OverrideCategory =
        BasicServiceGroupList/
            Index =
            BSG-Ext-BearerService =
            BSG-Ext-Teleservice =
    EMLPP-Info/
        MaximumEntitledPriority =
        DefaultPriority =
    ODB-Data/
        ODB-GeneralData =
        ODB-HPLMN-Data =
    RegionalSubscriptionData/
        Index =
        RegionalSubscriptionData =
    VBSSubscriptionData/
        Index =
        VBS-GroupId =
        BroadcastInitEntitlement =
    VGCSSubscriptionData/
        Index =
        VGCS-GroupId =
        AdditionalSubscriptions =
        AdditionalInfo =
        LongGroupId =
    LCSInformation/
        GMLC-List/
            Index =
            GMLC =

```

```

LCS-PrivacyExceptionList/
  Index =
  PE-SS-Code =
  SS-Status =
  LCSNotificationToMSUser =
  ExternalClientList/
    Index =
    ClientIdentity.ExternalAddress =
    ExtCliGMLC-Restriction =
    ExtCliNotificationToMSUser =
  PLMNClientList/
    Index =
    PLMNClient =
  ServiceTypeList/
    Index =
    ServiceTypeIdentity =
    SerTypeGMLC-Restriction =
    SerTypeNotificationToMSUser =
MOLR-List/
  Index =
  MOLR-SS-Code =
  MOLR-SS-Status =
MC-SS-Info/
  MC-SS-Code =
  MC-SS-Status =
  NbrSB =
  NbrUser =
SGSN-CAMEL-SubscriptionInfo/
  GPRS-CSI/
    GPRS-CamelCapabilityHandling =
    GPRS-NotificationToCSE =
    GPRS-CSI-Active =
    GPRS-CamelTDPDataList/
      Index =
      GPRS-TriggerDetectionPoint =
      GPRS-ServiceKey =
      GPRS-GSMSCF-Address =
      DefaultSessionHandling =
  MO-SMS-CSI/
    MOSMS-CamelCapabilityHandling =
    MOSMS-NotificationToCSE =
    MOSMS-CSI-Active =
    SMS-CAMEL-TDP-DataList/
      Index =
      MO-SMS-TriggerDetectionPoint =
      MO-ServiceKey =
      MO-GSMSCF-Address =
      MO-DefaultSMShandling =
  MT-SMS-CSI/
    MTSMS-CamelCapabilityHandling =
    MTSMS-NotificationToCSE =
    MTSMS-CSI-Active =
    SMS-CAMEL-TDP-DataList/
      Index =
      MT-SMS-TriggerDetectionPoint =
      MT-ServiceKey =
      MT-GSMSCF-Address =
      MT-DefaultSMShandling =
  MT-SMSCAMELTDP-CriteriaList/
    Index =
    SMS-TriggerDetectionPoint =
    TPDU-TypeCriterion =
MG-CSI/
  MobilityTriggers =

```

```

MG-ServiceKey =
MG-GSMSCF-Address =
MG-NotificationToCSE =
MG-CSI-Active =
ProfileMappings/
  imsi = 100,Profile1
  naea = 20,Profile2
  naea = 30,Profile3

[ //localhost/Radius/Profiles ]
Entries 1 to 6 from 6 total entries
Current filter: <all>

default-PPP-users/
default-SLIP-users/
default-Telnet-users/
Profile1/
Profile2/
Profile3/

```

Table 14-2 shows the restore data mapping parameters.

**Table 14-2** Restore Data Mappings and Profile Mappings Parameters

Parameter	Description
IMSI	IMSI received in the response from HLR.
Naea-Preferred CI	North American Equal Access preferred Carrier ID List. A list of the preferred carrier identity codes that are subscribed to.
Roaming Restricted In Sgsn Due To Unsupported Feature	Indicates that a subscriber is not allowed to roam in the current Service GPRS Support Node (SGSN) or Cisco Mobility Management Entity (MME) area.
Network Access Mode	The Network Access Mode (NAM) defines if the subscriber is registered to get access to the CS (non-GPRS/EPS network), to the PS (GPRS/EPS) network or to both networks. NAM describes the first level of the subscriber data pseudo-tree below the IMSIroot. It is permanent subscriber data stored in the HSS / HLR and the SGSN with the Gs interface option, and the MME with the SGs interface option.
LMU Indicator	Indicates the presence of an LMU.
IST Alert Timer	Indicates the IST alert timer value that must be used in the Mobile Switching Center (MSC) to inform the HLR about the call activities that the subscriber performs.
Super Charger Supported In HLR	Indicates whether super charger concept is supported in HLR.
CS Allocation Retention Priority	Allocation-retention priority for Circuit Switched (CS). This parameter specifies relative importance to compare with other bearers about allocation and retention of bearer.
ChargingCharacteristics	Subscribed charging characteristics.

**Table 14-2 Restore Data Mappings and Profile Mappings Parameters (continued)**

<b>Parameter</b>	<b>Description</b>
Access Restriction Data	Allowed Recipient Access Table (RAT) according to subscription data.
UE Reachability Request Indicator	Indicates that the Home Subscriber Server (HSS) is awaiting a notification of user equipment (UE) reachability.
Category	Calling party category
LSA Information	These parameters refer to one or more localized service areas (LSAs) a subscriber may be a member of, together with the priority, the preferential access indicator, the active mode support indicator and active mode indication of each localized service area. The access right outside these localized service areas is also indicated.
<b>Subscriber Data</b>	
MSISDN	MSISDN value in the subscriber data.
Subscriber Status	Barring status of the subscriber, which could be Service Granted or Operator Determined Barring.
Roaming Restriction Due To Unsupported Feature	Indicates that the subscriber is not allowed to roam in the current MSC area.
Bearer Service List	List of extensible bearer services subscribed. Configure the index value to fetch only the required bearer services.
TeleService List	List of extensible teleservices subscribed. Configure the index value to fetch only the required teleservices.
Provisioned SS	List of supplementary services provisioned. Configure the index value to fetch only the required supplementary services.
ODB-Data	Operator Determined Barring (ODB) general data and ODB Home Public Land Mobile Network (HPLMN) specific data.
Regional Subscription Data	List of regional subscription areas (zones) in which the subscriber is allowed to roam. Configure the index value to fetch only the required zones.
VBS Subscription Data	List of Voice Broadcast Services (VBS) subscribed. Configure the index value to fetch only the required VBS.
VGCS Subscription Data	List of Voice Group Call Services (VGCS) subscribed. Configure the index value to fetch only the required VGCS.
<b>LCS Information</b>	
Live Communication Server (LCS) related information for the subscriber.	
GMLC-List	List of Gateway Mobile Location Centers (GMLCs) that are permitted to issue a call/session unrelated or call/session related MT-LR request. Configure the index value to fetch only the required GMLCs.
LCS-Privacy Exception List	Classes of LCS client that are allowed to locate any target Mobile Station (MS). Configure the index value to fetch only the required classes.

**Table 14-2 Restore Data Mappings and Profile Mappings Parameters (continued)**

Parameter	Description
MOLR-List	Code and status of Mobile Originating Location Request (MO-LR) subscribed. Configure the index value to fetch only the required requests.
MC-SS-Info	Parameters identifying Multicall (MC) supplementary services (SS).
SGSN-CAMEL-Subscription Info	Parameters identifying the subscribers as having Customized Application for Mobile Enhanced Logic (CAMEL) services that are invoked in the SGSN.
<b>ProfileMappings</b>	
Attribute	The RADIUS attribute to map the fetched profile data.
Value:Profile	Value of the attribute.

### Configuring Environment Variables to Fetch Subscriber Data Values

You can configure an environment variable to fetch the required values from the subscriber data packets. You can run a script to fetch the environment variable along with the values. See the example below:

```
proc FetchBearerService {request response environ} {
    set bearerService [ $environ get bs-ext ]
    $request trace 2 "BearerService value fetched is " $bearerService
}
```

In the above script bs-ext is the environment variable that is configured. If the values fetched from BearerServiceList are 17,18,19,20 and 21, the above script returns the value 17:18:19:20:21.

Similarly we can run scripts to retrieve other environment variables as well.

## Blacklisting Support for SIGTRAN-M3UA Remote Server

Prime Access Registrar supports blacklisting of IMSI or IP address values for SIGTRAN-M3UA remote servers.

You can configure a SIGTRAN-M3UA remote server with EAP-SIM or EAP-AKA service, and then choose to configure blacklisting as part of the global title translation script of the remote server. For more information about blacklisting, see the “Using Extension Points” chapter of the [Cisco Prime Access Registrar 8.0 Administrator Guide](#).

## Support for SCTP Multihoming in SIGTRAN-M3UA

Stream Control Transmission Protocol (SCTP) is an IP transport protocol that supports data exchange between exactly two endpoints. Multihoming feature of SCTP provides the ability for a single SCTP endpoint to support multiple IP addresses. With this feature, each of the two endpoints during an SCTP association can specify multiple points of attachment. Each endpoint will be able to receive messages from any of the addresses associated with the other endpoint. With the use of multiple interfaces, data can be sent to alternate addresses when failures occur and thus Prime Access Registrar runs successfully even during network failures.

Prime Access Registrar allows you to configure multiple source and destination addresses on the remote server. The following shows an example configuration of SIGTRAN-M3UA remote server with multiple source and destination addresses:

```
[ /Radius/RemoteServers/m3ua ]
  Name = m3ua
  Description =
  Protocol = sigtran-m3ua
  SourcePort = 2805
  LocalSubSystemNumber = 149
  DestinationPort = 2855
  IMSITranslationScript~ =
  GlobalTitleTranslationScript~ =
  Timeout = 15
  ReactivateTimerInterval = 300000
  LimitOutstandingRequests = FALSE
  MaxOutstandingRequests = 0
  MAPVersion = 3
  NetworkVariant = ITU
  SubServiceField = NAT
  TCAPVariant = ITU96
  NetworkAppearance = 1
  NetworkIndicator = NAT
  MLCNumber = 123456789012345
  TrafficMode = LOADSHARE
  LoadShareMode = SLS
  RoutingIndicator = RTE_SSN
  RoutingParameters/
  OriginPointCode = 2
  DestinationPointCode = 4
  RemoteSubSystemNumber = 6
  OPCMask = 16383
  DPCMask = 16383
  ServiceIndicatorOctet = 0
  RoutingContext = 11
  SourceIPAddresses/
  DestinationIPAddresses/
--> cd SourceIPAddresses
--> add 1 192.168.0.2
--> add 2 192.168.0.3
--> cd ../DestinationIPAddresses
--> add 1 192.168.0.5
--> add 2 192.168.0.6
```

In the above example, the link between IP addresses 192.168.0.2 and 192.168.0.5 acts as the primary link and the link between IP addresses 192.168.0.3 and 192.168.0.6 acts as the secondary link. With the Multihoming feature, if one of the interfaces in the primary link is down, the secondary link carries the active traffic. On restoration of the IP address, the traffic switches back to the primary link.

## Tuning Global SIGTRAN Parameters

Prime Access Registrar provides a CLI tool SigtranXMLEdit that allows you to edit the values of the global SIGTRAN XML parameters. The tool is available under the <installation directory>/bin directory, e.g. /cisco-ar/bin and the parameters are available in the *default.xml* file under the /cisco-ar/m3ua-cfg directory.

[Table 14-3](#) lists the global SIGTRAN parameters that you can edit using the CLI tool.

**Table 14-3 Global SIGTRAN Parameter**

Parameter	Description
rtoMin	Minimum value of retransmission timeout
rtoMax	Maximum value of retransmission timeout
rtoInitial	Initial value of retransmission timeout
alpha	Retransmission timeout alpha value
beta	Retransmission timeout beta value
maxAssocReTx	Maximum association retransmission
maxPathReTx	Maximum path retransmission
maxInitReTx	Maximum initial retransmission
cookieLife	Cookie life
intervalTm	Heartbeat interval
maxAckDelayTm	SACK period
maxNmbInStrms	Maximum number of inbound streams
maxNmbOutStrms	Maximum number of outbound streams
mtuInitial	Initial value of maximum transmission unit
mtuMinInitial	Minimum Initial value of maximum transmission unit
mtuMaxInitial	Maximum Initial value of maximum transmission unit

To edit the SIGTRAN parameters:

**Step 1** Launch the CLI tool **SIGTRANXMLeDit** from the `/cisco-ar/bin` directory.

The tool displays the list of editable parameters available in the `default.xml` file as shown below.

1. RTO min (RTOMI) from the header `_sbSctSapCfg`
2. RTO max (RTOMA) from the header `_sbSctSapCfg`
3. RTO Initial (RTOI) from the header `_sbSctSapCfg`
4. RTO Alpha (RTOA) from the header `_sbGenReCfg`
5. RTO Beta (RTOB) from the header `_sbGenReCfg`
6. Assoc. Max retrans. (AMR) from the header `_sbGenReCfg`
7. Path Max retrans. (PMR) from the header `_sbGenReCfg`
8. Initial retrans. Attempts (IMR) from the header `_sbGenReCfg`
9. Cookie life (VCL) from the header `_sbSctSapReCfg`
10. HB interval (HBI) from the header `_sbSctSapReCfg`
11. SACK period (TSACK) from the header `_sbSctSapReCfg`
12. Streams per association (MIS/MOS) from the header `_sbGenCfg`
13. Maximum Transmission unit (MTU) from the header `_sbGenCfg`

The tool prompts you to enter the new value against the first parameter as shown below.

```
Enter values for the following parameters (just press 'return' to skip):
Maximum number of inbound streams "maxNmbInStrms" [ 1024 ] :
```

In this example, 1024 is the value that exists for the parameter in the `default.xml` file.

**Step 2** Type the new value and press ENTER or just press ENTER to skip and proceed to the next parameter. Perform this step for all parameters as shown below.

```
Enter values for the following parameters (just press 'return' to skip):
```

```

Maximum number of inbound streams "maxNmbInStrms" [ 1024 ] : 87
Maximum number of outbound streams "maxNmbOutStrms" [ 1024 ] : 90
Initial value of Maximum Transmission Unit "mtuInitial" [ 1500 ] :
Minimum Initial value of Maximum Transmission Unit "mtuMinInitial" [ 1500 ] : 65
Maximum Initial value of Maximum Transmission Unit "mtuMaxInitial" [ 1500 ] : 33
Maximum initial Retransmission "maxInitReTx" [ 5 ] : 9
Maximum association Retransmission "maxAssocReTx" [ 10 ] : 4
Maximum path Retransmission "maxPathReTx" [ 5 ] : 2
Alpha value "alpha" [ 12 ] : 15
Beta value "beta" [ 25 ] : 34
MaxAckDelayTm "maxAckDelayTm" [ 2 ] : 89<br>
  Initial value of Retransmission timeout "rtoInitial" [ 15 ] :
Minimum value of Retransmission timeout "rtoMin" [ 15 ] :
Maximum value of Retransmission timeout "rtoMax" [ 60 ] :
CookieLife "cookieLife" [ 60 ] : 67
IntervalTm "intervalTm" [ 15 ] : 89
Do you want to save all the changes? [ Yes/No ]yes

```

**Step 3** When prompted for a confirmation, type **Yes** and press ENTER to save the changes. The tool displays the modified parameters with the new and old values.

```

Changed Value of maxNmbInStrms is 87 <- 1024
Changed Value of maxNmbOutStrms is 90 <- 1024
Changed Value of mtuMinInitial is 65 <- 1500
Changed Value of mtuMaxInitial is 33 <- 1500
Changed Value of maxInitReTx is 9 <- 5
Changed Value of maxAssocReTx is 4 <- 10
Changed Value of maxPathReTx is 2 <- 5
Changed Value of alpha is 15 <- 12
Changed Value of beta is 34 <- 25
Changed Value of maxAckDelayTm is 89 <- 2
Changed Value of cookieLife is 67 <- 60
Changed Value of intervalTm is 89 <- 15
[root@ar-lnx-vm061 bin]#

```

## SIGTRAN-M3UA Logs

The following logs are applicable for SIGTRAN-M3UA:

- stack.log—Logs the interaction between Prime Access Registrar and STP/HLR.
- sm.log—Logs the internal debug information for SIGTRAN-M3UA stack manager.
- m3ua.log—Logs the inter-process communication between Prime Access Registrar and SIGTRAN-M3UA stack.
- cliActivity.log—Logs the initialization and command interactions.



## Using SNMP

---

This chapter provides the following information about Cisco Prime Access Registrar (Prime Access Registrar) support for SNMP:

- [Overview, page 15-1](#)
- [Supported MIBs, page 15-1](#)
- [SNMP Traps, page 15-3](#)
- [SNMP Version 3 Support, page 15-13](#)

### Overview

Prime Access Registrar provides SNMP MIB and trap support for users of network management systems. The supported MIBs enable the network management station to collect state and statistic information from an Prime Access Registrar server. The traps enable Prime Access Registrar to notify interested network management stations of failure or impending failure conditions.

Prime Access Registrar supports the MIBs defined in the following RFCs:

- RADIUS Authentication Client MIB for IPv6, RFC 4668
- RADIUS Authentication Server MIB for IPv6, RFC 4669
- RADIUS Accounting Client MIB for IPv6, RFC 4670
- RADIUS Accounting Server MIB for IPv6, RFC 4671
- CISCO Diameter Base Protocol MIB

Prime Access Registrar MIB support enables a standard SNMP management station to check the current state of the server as well as the statistics on each client or each proxied remote server.

Prime Access Registrar Trap support enables a standard SNMP management station to receive trap messages from an Prime Access Registrar server. These messages contain information indicating that either the server was brought up or down, or that the proxied remote server is down or has come back online.

### Supported MIBs

The MIBs supported by Prime Access Registrar enable a standard SNMP management station to check the current state of the server and statistics for each client or proxied remote server.

This section contains the following topics:

- [RADIUS-AUTH-CLIENT-MIB](#)
- [RADIUS-AUTH-SERVER-MIB](#)
- [RADIUS-ACC-CLIENT-MIB](#)
- [RADIUS-ACC-SERVER-MIB](#)
- [CISCO-DIAMETER-BASE-PROTOCOL-MIB](#)
- [Diameter SNMP and Statistics Support](#)
- [TACACS+ SNMP and Statistics Support](#)

## RADIUS-AUTH-CLIENT-MIB

The RADIUS-AUTH-CLIENT-MIB describes the client side of the RADIUS authentication protocol. The information contained in this MIB is useful when an Prime Access Registrar server is used as a proxy server.

## RADIUS-AUTH-SERVER-MIB

The RADIUS-AUTH-SERVER-MIB describes the server side of the RADIUS authentication protocol. The information contained in this MIB describes managed objects used for managing a RADIUS authentication server.

## RADIUS-ACC-CLIENT-MIB

The RADIUS-ACC-CLIENT-MIB describes the client side of the RADIUS accounting protocol. The information contained in this MIB is useful when an Prime Access Registrar server is used for accounting.

## RADIUS-ACC-SERVER-MIB

The RADIUS-ACC-CLIENT-MIB describes the server side of the RADIUS accounting protocol. The information contained in this MIB is useful when an Prime Access Registrar server is used for accounting.

## CISCO-DIAMETER-BASE-PROTOCOL-MIB

Prime Access Registrar uses the CISCO-DIAMETER-BASE-PROTOCOL-MIB as an interface to query the Diameter statistics, though configuring the Diameter through SNMP is not possible. Prime Access Registrar supports LocalStatistics and PeerStatistics only. The LocalStats provides statistical information about the local diameter server and the PeerStats provides statistical information about the peers and the messages to/from the peers.

## Diameter SNMP and Statistics Support

Prime Access Registrar also supports Diameter SNMP MIB (CISCO-DIAMETER-BASE-PROTOCOL-MIB) to describe the Diameter Base Protocol statistics.

Prime Access Registrar supports statistic of Diameter messages to include the additional counters. This is supported through the CLI/GUI and SNMP. The diameter statistics includes peer statistics and global summary statistics details.

## TACACS+ SNMP and Statistics Support

Prime Access Registrar supports the CISCO-AAA-SERVER-MIB to describe the statistics of TACACS+ protocol. TACACS+ protocol is used to authenticate an user via various services such as login services. This is supported through the CLI/GUI and SNMP.

## SNMP Traps

The traps supported by Prime Access Registrar enable a standard SNMP management station to receive trap messages from an Prime Access Registrar server. These messages contain information indicating whether a server was brought up or down, or that the proxied remote server is down or has come back online.

A trap is a network message of a specific format issued by an SNMP entity on behalf of a network management agent application. A trap is used to provide the management station with an asynchronous notification of an event.

When a trap is generated, a single copy of the trap is transmitted as a trap PDU to each destination contained within a list of trap recipients.

The list of trap recipients is shared by all events and is determined at server initialization time along with other trap configuration information. The list of trap recipients dictates where Prime Access Registrar traps are directed.

The configuration of any other SNMP agent on the host is ignored. By default, all traps are enabled but no trap recipients are defined. By default, no trap is sent until trap recipients are defined.

Traps are configured using the command line interface (CLI). After configuring traps, the configuration information is re initialized when a server reload or restart occurs.



### Note

---

SNMP queries and traps communication can be performed over IPv6.

---

When you configure traps, you must provide the following information:

- List of trap recipients (community string for each)
- Suppressing traps for any type of message
- Frequency of traps for any type of message

This topic contains the following sections:

- [Supported Traps, page 15-4](#)
- [Configuring Traps, page 15-12](#)

- [Community String](#), page 15-12

## Supported Traps

The traps supported by Prime Access Registrar enable the Prime Access Registrar server to notify interested management stations of events, failure, or impending failure conditions. Traps are a network message of a specific format issued by an SNMP entity on behalf of a network management agent application. Traps are used to provide the management station with an asynchronous notification of an event.

This section contains the following topics:

- [carServerStart](#)
- [carServerStop](#)
- [carInputQueueFull](#)
- [carInputQueueNotVeryFull](#)
- [carDiaInputQueueFull](#)
- [carDiaInputQueueNotFull](#)
- [carOtherAuthServerNotResponding](#)
- [carOtherAuthServerResponding](#)
- [carOtherAccServerNotResponding](#)
- [carOtherAccServerResponding](#)
- [carAccountingLoggingFailure](#)
- [carLicenseUsage](#)
- [carSigtranLicenseUsage](#)
- [carDiameterPeerDown](#)
- [carDiameterPeerUp](#)
- [carTPSCapacityFull](#)
- [carTPSCapacityNotFull](#)
- [carSigtranTPSCapacityFull](#)
- [carSigtranTPSCapacityNotFull](#)
- [carSessionCapacityFull](#)
- [carSessionCapacityNotFull](#)
- [carSigtranSessionCapacityFull](#)
- [carSigtranSessionCapacityNotFull](#)
- [carLicenseUsageReset](#)
- [carSigtranLicenseUsageReset](#)
- [carReplicationSyncFailure](#)
- [carReplicationSuccess](#)
- [TLSCClientConnectionUpTrap](#)
- [TLSCClientConnectionClosedTrap](#)

## carServerStart

**carServerStart** signifies that the server has started on the host from which this notification was sent. This trap has one object, *carNotifStartType*, which indicates the start type. A *firstStart* indicates this is the server process' first start. *reload* indicates this server process has an internal reload. This typically occurs after rereading some configuration changes, but *reload* indicates this server process did not quit during the reload process.

## carServerStop

**carServerStop** signifies that the server has stopped normally on the host from which this notification was sent.

## carInputQueueFull

**carInputQueueFull** indicates that the percentage of use of the packet input queue has reached its high threshold. This trap has two objects:

- *carNotifInputQueueHighThreshold*—indicates the high limit percentage of input queue usage
- *carNotifInputQueueLowThreshold*—indicates the low limit percentage of input queue usage

By default, *carNotifInputQueueHighThreshold* is set to 90% and *carNotifInputQueueLowThreshold* is set to 60%.

**Note**

---

The values for these objects cannot be changed at this time. You will be able to modify them in a future release of Prime Access Registrar.

---

After this notification has been sent, another notification of this type will not be sent again until the percentage usage of the input queue goes below the low threshold.

If the percentage usage reaches 100%, successive requests might be dropped, and the server might stop responding to client requests until the queue drops down again.

## carInputQueueNotVeryFull

**carInputQueueNotVeryFull** indicates that the percentage usage of the packet input queue has dropped below the low threshold defined in *carNotifInputQueueLowThreshold*. This trap has two objects:

- *carNotifInputQueueHighThreshold*—indicates the high limit percentage of input queue usage
- *carNotifInputQueueLowThreshold*—indicates the low limit percentage of input queue usage

After this type of notification has been sent, it will not be sent again until the percentage usage goes back up above the high threshold defined in *carNotifInputQueueHighThreshold*.

## carDiaInputQueueFull

**carDiaInputQueueFull** signifies that the percentage of use of the Diameter packet input queue has reached its high threshold. This trap has two objects:

- *carNotifDiaInputQueueHighThreshold*—indicates the high limit percentage of Diameter packet input queue usage.

- *carNotifDiaInputQueueLowThreshold*—indicates the low limit percentage of the Diameter packet input queue usage.

If the percentage usage reaches 100%, successive request is dropped, and the server stops responding to client requests until the queue drops down again. After this notification is sent, this type of notification will not be sent again until the percentage usage of the input queue goes back down below the low threshold.

## carDiaInputQueueNotFull

**carDiaInputQueueNotFull** signifies that the percentage of use of Diameter packet input queue has dropped below the low threshold defined in *carNotifDiaInputQueueLowThreshold*. This trap has two objects:

- *carNotifDiaInputQueueHighThreshold*—indicates the high limit percentage of Diameter packet input queue usage.
- *carNotifDiaInputQueueLowThreshold*—indicates the low limit percentage of the Diameter packet input queue usage.

After this type of notification has been sent, it will not be sent again until the percentage usage goes back up above the high threshold defined in *carNotifDiaInputQueueHighThreshold*.

## carOtherAuthServerNotResponding

**carOtherAuthServerNotResponding** indicates that an authentication server is not responding to a request sent from this server. This trap has three objects:

- *radiusAuthServerAddress*—indicates the identity of the concerned server
- *radiusAuthClientServerPortNumber*—indicates the port number of the concerned server
- *carAuthServerType*—indicates the type of the concerned server

The index of these three objects identifies the entry in *radiusAuthServerTable* and *carAccServerExtTable* which maintains the characteristics of the concerned server.



### Note

One should not rely solely on **carOtherAuthServerNotResponding** for server state. Several conditions, including a restart of the Prime Access Registrar server, could result in either multiple *carOtherAuthServerNotResponding* notifications being sent or in a *carOtherAuthServerResponding* notification *not* being sent. NMS can query the *carAuthServerRunningState* in *carAuthServerExtTable* for the current running state of this server.

## carOtherAuthServerResponding

**carOtherAuthServerResponding** signifies that an authentication server which had formerly been in a *down* state is now responding to requests from the Prime Access Registrar server. This trap has three objects:

- *radiusAuthServerAddress*—indicates the identity of the concerned server
- *radiusAuthClientServerPortNumber*—indicates the port number of the concerned server
- *carAuthServerType*—indicates the type of the concerned server

The index of these three objects identifies the entry in *radiusAuthServerTable* and *carAccServerExtTable* which maintains the characteristics of the concerned server.

One should not rely on receiving this notification as an indication that all is well with the network. Several conditions, including a restart of the Prime Access Registrar server, could result in either multiple *carOtherAuthServerNotResponding* notifications being sent or in a *carOtherAuthServerResponding* notification *not* being sent. The NMS can query the *carAuthServerRunningState* in *carAuthServerExtTable* for the current running state of this server.

## carOtherAccServerNotResponding

**carOtherAuthServerNotResponding** signifies that an accounting server is not responding to the requests sent from this server. This trap has three objects:

- *radiusAccServerAddress*—indicates the identity of the concerned server
- *radiusAccClientServerPortNumber*—indicates the port number of the concerned server
- *carAccServerType*—indicates the type of the concerned server

The index of these three objects identifies the entry in *radiusAuthServerTable* and *arAccServerExtTable* which maintains the characteristics of the concerned server.

One should not solely rely on this for server state. Several conditions, including the restart of the Prime Access Registrar server, could result in either multiple *carOtherAccServerNotResponding* notifications being sent or in a *carOtherAccServerResponding* notification *not* being sent. The NMS can query the *carAccServerRunningState* in *carAccServerExtTable* for current running state of this server.

## carOtherAccServerResponding

**carOtherAccServerResponding** signifies that an accounting server that had previously sent a *not responding* message is now responding to requests from the Prime Access Registrar server. This trap has three objects:

- *radiusAccServerAddress*—indicates the identity of the concerned server
- *radiusAccClientServerPortNumber*—indicates the port number of the concerned server
- *carAccServerType*—indicates the type of the concerned server

The index of these three objects identifies the entry in *radiusAuthServerTable* and *arAccServerExtTable* which maintains the characteristics of the concerned server.

One should not rely on the reception of this notification as an indication that all is well with the network. Several conditions, including the restart of the Prime Access Registrar server, could result in either multiple *carOtherAccServerNotResponding* notifications being sent or in a **carOtherAccServerResponding** notification *not* being sent. The NMS can query the *carAccServerRunningState* in *carAccServerExtTable* for the current running state of this server.

## carAccountingLoggingFailure

**carAccountingLoggingFailure** signifies that this Prime Access Registrar server cannot record accounting packets locally. This trap has two objects:

- *carNotifAcctLogErrorReason*—indicates the reason packets cannot be recorded locally
- *carNotifAcctLogErrorInterval*—indicates how long to wait until another notification of this type might be sent. A value of 0 (zero) indicates no time interval checking, meaning that no new notification can be sent until the error condition is corrected.

## carLicenseUsage

**carLicenseUsage** signifies the percentage of transactions per second (TPS) usage or session usage from the available license values.

### TPS

The TPS trap is generated when the Prime Access Registrar server reaches license usage slabs namely 80%, 90%, 100%, and 110%. These traps are generated only once for every slab during the increasing steady state. Increasing steady state is a state when Prime Access Registrars' incoming request rate shows 80% of the license usage over a period of 20 minutes. These traps will be regenerated only if a increasing steady state is observed after a decreasing steady state.

### Concurrent Session

The concurrent session trap is generated when the Prime Access Registrar server reaches 80%. The incoming traffic slabs defined for trap generation are 80%, 90%, 100%, and 110% of the licensed Concurrent Sessions. These traps are generated once for every slab during the increasing steady state.

## carSigtranLicenseUsage

**carSigtranLicenseUsageTrap** signifies the percentage of SIGTRAN TPS usage or SIGTRAN session usage from the available license values.

## carDiameterPeerDown

**carDiameterPeerDown** signifies that a Diameter peer is down. The identity of the peer is given by `cdbpPeerIpAddress`.

## carDiameterPeerUp

**carDiameterPeerUp** signifies that a Diameter peer is up. The identity of the peer is given by `cdbpPeerIpAddress`.

## carTPSCapacityFull

**carTPSCapacityFull** signifies that the TPS of the Prime Access Registrar server has reached the configured high threshold capacity. This trap has the following objects:

- *carNotifTPSHighThreshold*—indicates the maximum limit of the TPS of the Prime Access Registrar server.
- *carNotifTPSLowThreshold*—indicates the minimum limit of the TPS of the Prime Access Registrar server.
- *carServerTPSUsage*—indicates the current TPS usage of the Prime Access Registrar server.

After this notification is sent, this type of notification will not be sent again until the TPS of Prime Access Registrar server reduces below the configured *carNotifTPSLowThreshold* value.

## carTPSCapacityNotFull

**carTPSCapacityNotFull** signifies that the TPS of the Prime Access Registrar server has dropped below the configured low threshold capacity. This trap has the following objects:

- *carNotifTPSHighThreshold*—indicates the maximum limit of the TPS of the Prime Access Registrar server.
- *carNotifTPSLowThreshold*—indicates that the minimum limit of the TPS of the Prime Access Registrar server.
- *carServerTPSUsage*—indicates the current TPS usage of the Prime Access Registrar server.

After this notification is sent, this type of notification will not be sent again until the TPS of Prime Access Registrar server increases beyond the configured *carNotifTPSHighThreshold* value.

## carSigtranTPSCapacityFull

**carSigtranTPSCapacityFull** signifies that the SIGTRAN TPS of the Prime Access Registrar server has reached the configured high threshold capacity. This trap has the following objects:

- *carNotifSigtranTPSHighThreshold*—indicates the maximum limit of the SIGTRAN TPS of the Prime Access Registrar server.
- *carNotifSigtranTPSLowThreshold*—indicates the minimum limit of the SIGTRAN TPS of the Prime Access Registrar server.
- *carServerSigtranTPSUsage*—indicates the current SIGTRAN TPS usage of the Prime Access Registrar server.

After this notification is sent, this type of notification will not be sent again until the SIGTRAN TPS of Prime Access Registrar server reduces below the configured *carNotifSigtranTPSLowThreshold* value.

## carSigtranTPSCapacityNotFull

**carSigtranTPSCapacityNotFull** signifies that the SIGTRAN TPS of the Prime Access Registrar server has reached the configured low threshold capacity. This trap has the following objects:

- *carNotifSigtranTPSHighThreshold*—indicates the maximum limit of the SIGTRAN TPS of the Prime Access Registrar server.
- *carNotifSigtranTPSLowThreshold*—indicates the minimum limit of the SIGTRAN TPS of the Prime Access Registrar server.
- *carServerSigtranTPSUsage*—indicates the current SIGTRAN TPS usage of the Prime Access Registrar server.

After this notification is sent, this type of notification will not be sent again until the SIGTRAN TPS of Prime Access Registrar server increases beyond the configured *carNotifSigtranTPSHighThreshold* value.

## carSessionCapacityFull

**carSessionCapacityFull** signifies that the session TPS of the Prime Access Registrar server has reached the configured high threshold capacity. This trap has the following objects:

- *carNotifSessionHighThreshold*—indicates the maximum limit of the session TPS of the Prime Access Registrar server.

- *carNotifSessionLowThreshold*—indicates the minimum limit of the session TPS of the Prime Access Registrar server.
- *carServerSessionUsage*—indicates the current session TPS usage of the Prime Access Registrar server.

After this notification is sent, this type of notification will not be sent again until the session TPS of Prime Access Registrar server reduces below the configured *carNotifSessionLowThreshold* value.

## carSessionCapacityNotFull

**carSessionCapacityNotFull** signifies that the session TPS of the Prime Access Registrar server has reached the configured low threshold capacity. This trap has the following objects:

- *carNotifSessionHighThreshold*—indicates the maximum limit of the session TPS of the Prime Access Registrar server.
- *carNotifSessionLowThreshold*—indicates the minimum limit of the session TPS of the Prime Access Registrar server.
- *carServerSessionUsage*—indicates the current session TPS usage of the Prime Access Registrar server.

After this notification is sent, this type of notification will not be sent again until the session TPS of Prime Access Registrar server increases beyond the configured *carNotifSessionHighThreshold* value.

## carSigtranSessionCapacityFull

**carSigtranSessionCapacityFull** signifies that the SIGTRAN session TPS of the Prime Access Registrar server has reached the configured high threshold capacity. This trap has the following objects:

- *carNotifSigtranSessionHighThreshold*—indicates the maximum limit of the SIGTRAN session TPS of the Prime Access Registrar server.
- *carNotifSigtranSessionLowThreshold*—indicates the minimum limit of the SIGTRAN session TPS of the Prime Access Registrar server.
- *carServerSessionUsage*—indicates the current SIGTRAN session TPS usage of the Prime Access Registrar server.

After this notification is sent, this type of notification will not be sent again until the SIGTRAN session TPS of Prime Access Registrar server reduces below the configured *carNotifSigtranSessionLowThreshold* value.

## carSigtranSessionCapacityNotFull

**carSigtranSessionCapacityNotFull** signifies that the SIGTRAN session TPS of the Prime Access Registrar server has reached the configured low threshold capacity. This trap has the following objects:

- *carNotifSigtranSessionHighThreshold*—indicates the maximum limit of the SIGTRAN session TPS of the Prime Access Registrar server.
- *carNotifSigtranSessionLowThreshold*—indicates the minimum limit of the SIGTRAN session TPS of the Prime Access Registrar server.
- *carServerSessionUsage*—indicates the current SIGTRAN session TPS usage of the Prime Access Registrar server.

After this notification is sent, this type of notification will not be sent again until the SIGTRAN session TPS of Prime Access Registrar server increases beyond the configured *carNotifSigtranSessionHighThreshold* value.

## carLicenseUsageReset

**carLicenseUsageReset** signifies that the server usage is nominal after exceeding the license thresholds. This notification carries the percentage of License Usage.

## carSigtranLicenseUsageReset

**carSigtranLicenseUsageReset** signifies that server SIGTRAN usage is nominal after exceeding the license thresholds. This notification carries the percentage of SIGTRAN License Usage.

## carReplicationSyncFailure

**carReplicationSyncFailure** notifies that there is a synchronization failure in Prime Access Registrar replication. This notification is triggered when there is a failure in sync message exchanges or upon out of sync configuration detection. This trap has four objects:

- *carNotifReplicationMasterInetAddrType*—indicates the type of Internet address of the Master, which could be IPv4 address, IPv6 address, or DNS domain name.
- *carNotifReplicationMasterIPAddress*—indicates the IP address of the Master referred to using the version-neutral IP address.
- *carNotifReplicationMemberIPAddress*—indicates the type of Internet address of the Member, which could be IPv4 address, IPv6 address, or DNS domain name.
- *carNotifReplicationMemberInetAddrType*—indicates the IP address of the Member referred to using the version-neutral IP address.

## TLSClientConnectionUpTrap

**TLSClientConnectionUpTrap** is sent from Prime Access Registrar when first connection is established by a RADIUS-TLS client.

## TLSClientConnectionClosedTrap

**TLSClientConnectionClosedTrap** is sent from Prime Access Registrar when all the connections are closed by the RADIUS-TLS client.

## carReplicationSuccess

**carReplicationSuccess** notifies that replication synchronization, which had formerly been in a down state is now resolved. This trap has four objects:

- *carNotifReplicationMasterInetAddrType*—indicates the type of Internet address of the Master, which could be IPv4 address, IPv6 address, or DNS domain name.
- *carNotifReplicationMasterIPAddress*—indicates the IP address of the Master referred to using the version-neutral IP address.

- `carNotifReplicationMemberIPAddress`—indicates the type of Internet address of the Member, which could be IPv4 address, IPv6 address, or DNS domain name.
- `carNotifReplicationMemberInetAddrType`—indicates the IP address of the Member referred to using the version-neutral IP address.

## Configuring Traps

The Prime Access Registrar SNMP implementation uses various configuration files to configure its applications.

This section contains the following topics:

- [SNMP Configuration](#)
- [Configuring Trap Recipient](#)

## SNMP Configuration

A sample configuration file is available in `/cisco-ar/ucd-snmp/share/snmp/snmpd.conf`. This configuration file is used to configure SNMP query permissions and trap recipients.

## Configuring Trap Recipient

The following example shows the default configuration that sets up trap recipients for SNMP versions v1 and v2c.



### Note

Most sites use a single NMS, not two as shown below.

```
# -----
trapcommunity trapcom
trapsink zubat trapcom 162
trap2sink ponyta trapcom 162
#####
```



### Note

**trapsink** is used in SNMP version 1; **trap2sink** is used in SNMP version 2.

**trapcommunity** defines the default community string to be used when sending traps. This command must appear prior to **trapsink** or **trap2sink** which use this community string.

**trapsink** and **trap2sink** are defined as follows:

```
trapsink  hostname  community  port
trap2sink hostname  community  port
```

## Community String

A community string is used to authenticate the trap message sender (SNMP agent) to the trap recipient (SNMP management station). A community string is required in the list of trap receivers.

# SNMP Version 3 Support

The SNMP Version 3 (SNMPv3) feature provides secure access to devices by authenticating and encrypting data packets over the network. SNMPv3 is an inter-operable, standards-based protocol. SNMPv3 is a security model in which an authentication strategy is set up for a user and the group in which the user resides. Security level is the permitted level of security within a security model. A combination of a security model and a security level determines which security mechanism is used when handling an SNMP packet.

The security features provided in SNMPv3 are:

- Message integrity—Ensures that a packet has not been tampered with during transit.
- Authentication—Determines that the message is from a valid source.
- Encryption—Scrambles the content of a packet to prevent it from being learned by an unauthorized source.

Table 15-1 lists the security levels supported by SNMPv3.

**Table 15-1 Security Levels Supported by SNMPv3**

Security Level	Authentication Mechanism	Encryption Mechanism	Description
noAuthNoPriv	With Username	None	Uses a username match for authentication.
authNoPriv	Message Digest Algorithm 5 (MD5) or Secure Hash Algorithm (SHA)	None	Provides authentication based on the Hashed Message Authentication Code (HMAC)-MD5 or HMAC-SHA algorithms.
authPriv	MD5 or SHA	Data Encryption Standard (DES) or Advanced Encryption Standard (AES)	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. In addition to authentication, provides encryption based on the DES/AES standards.

## Configuring SNMPv3 in Prime Access Registrar

This topic contains the following sections:

- [Prerequisites, page 15-13](#)
- [Creating Secure User for SNMP Query, page 15-14](#)
- [Configuring SNMPv3 Traps, page 15-14](#)

### Prerequisites

1. You must enable SNMP agent capability in Prime Access Registrar. To do so:

Log into the CLI. In SNMP object defined as /radius/advanced/snmp, set Enabled to true as shown below:

```
--> cd /radius/advanced/snmp/
[//localhost/Radius/Advanced/SNMP ]
Enabled = TRUE
TracingEnabled = FALSE
InputQueueHighThreshold = 90
InputQueueLowThreshold = 60
```

```
MasterAgentEnabled = TRUE
```

2. Make required changes to the `snmpd.conf` file located in `/cisco-ar/ucd-snmp/share/snmp`. After any change to `snmpd.conf`, restart the Prime Access Registrar server for the changes to take effect.

## Creating Secure User for SNMP Query

To use SNMPv3, you must define users with the prescribed security level and encryption methods in the following directory:

`/cisco-ar/ucd-snmp/share/snmp/snmpd.conf`

Table 15-2 provides sample commands for user creation.

**Table 15-2** User Creation in SNMPv3

Command Syntax	Sample Command	Description
<pre>createUser &lt;username&gt; &lt;authentication mechanism&gt; &lt;authentication password&gt; &lt;encryption mechanism&gt; &lt;encryption password&gt; rouser &lt;username&gt;</pre>	<pre>createUser securev3user SHA "snmpv3authPass" DES "snmpv3encPass" rouser securev3user</pre>	<p>Creates a user <i>securev3user</i> using SHA authentication mechanism and DES encryption protocol.</p> <p>This command is applicable for creating users with <code>authPriv</code> and <code>authNoPriv</code> security levels.</p>
<p><b>Note</b> We can use a combination of Authentication Algorithms (SHA and MD5) and Encryption algorithms (DES and AES) for creating user.</p>	<pre>createUser newuser rouser newuser noauth</pre>	<p>Creates a user <i>newuser</i> with <i>noauth</i> privilege level.</p> <p>This command is applicable for creating users with <code>noAuthNoPriv</code> security level.</p>

After modifying `snmpd.conf` file, ensure that you restart the Prime Access Registrar server for the changes to take effect.

## Configuring SNMPv3 Traps

When using SNMPv3, you must define the trap session command in the `snmpd.conf` file. Using this command, the required security definitions can be achieved as explained in Table 15-3.



**Note**

For receiving query responses and traps on the NMS, the NMS server must be configured corresponding to the definitions and configurations in `snmpd.conf` file of Prime Access Registrar.

**Table 15-3** *Trap Session Commands in SNMPv3*

Security Level	Description	Sample Trap Session Command	Command Action
noAuthNoPriv	Authentication using username and no encryption	trapsess -r 0 -v 3 -u snmpv3user -n "" -l noAuthNoPriv 10.10.10.11 162	Instructs SNMP agent to send traps to snmpv3 user to the defined NMS address
authNoPriv	Authentication using MD5 or SHA and no encryption	trapsess -r 0 -v 3 -u snmpv3user -n "" -l authNoPriv -a SHA -A snmpv3authPass -x AES -X snmpv3encPass 10.10.10.11 162  (or)  trapsess -r 0 -v 3 -u snmpv3user -n "" -l authNoPriv -a SHA -A snmpv3authPass 127.0.0.1 162	Instructs SNMP agent to send traps to snmpv3 user using SHA to the defined NMS address
authPriv	Authentication using MD5 or SHA and encryption using DES/AES	trapsess -r 0 -v 3 -u snmpv3user -n "" -l authPriv -a SHA -A snmpv3authPass -x AES -X snmpv3encPass 10.10.10.11 162	Instructs SNMP agent to send traps to snmpv3 user using SHA and AES algorithms to the defined NMS address

After modifying `snmpd.conf` file, ensure that you restart the Prime Access Registrar server for the changes to take effect.





## Backing Up the Database

---

This chapter describes the Cisco Prime Access Registrar (Prime Access Registrar) shadow backup facility, which ensures a consistent snapshot of Prime Access Registrar's database for backup purposes.

Because the Prime Access Registrar's database (called MCD) does a variety of memory caching, and might be active at any time, you cannot simply rely on doing system backups to protect the data in the database. At the time you run a system backup, there could be Prime Access Registrar operations in progress that cause the data copied to the system backup tape to be inconsistent and unusable as a replacement database.

To ensure a consistent backup, Prime Access Registrar uses a shadow backup facility. Once a day, at a configurable time, Prime Access Registrar suspends all activity to the database and takes a snapshot of the critical files. This snapshot is guaranteed to be a consistent view of the database, and it is preserved correctly on a system backup tape.

This chapter contains the following sections:

- [Configuration](#)
- [Recovery](#)
- [mcdshadow Command Files](#)

### Configuration

The only configuration for this facility is through a single entry in the system Registry at `$INSTALL/conf/car.conf` is the registry path to this item.

This entry is a string that represents the time-of-day at which the shadow backup is scheduled to occur (in 24 hour HH:MM format). The default is 12:45.

When you remove this entry or set it to an illegal value (for example, anything that does not begin with a digit), backups are suppressed.

### Command Line Utility

In addition to being available at a scheduled time of day, you can also force a shadow backup by using the `mcdshadow` utility located in the `$INSTALL/bin` directory. There are no command-line arguments.

This might take a few minutes to complete as a full copy of the database is created.

# Recovery

When it is necessary to use the shadow backup to recover data, either because the regular working database has been corrupted by a system crash, or because the disk on which it resides has become corrupted.

## Recovering the data using shadow backup

To use the shadow backup to recover data:

- 
- Step 1** Stop all Prime Access Registrar servers.
- Step 2** Make sure three files (**mcddb.d01**, **mcddb.d02**, and **mcddb.d03**) exist in the **\$INSTALL/data/db.bak** directory.
- Step 3** Copy the files into the **\$INSTALL/data/db** directory. Do not move them because they might be needed again.
- Step 4** Change directory to the **\$INSTALL/data/db** directory.
- ```
cd $INSTALL/data/db
```
- Step 5** Rebuild the key files by entering the command:
- ```
$INSTALL/bin/keybuild mcddb
```
- This might take several minutes.
- Step 6** As a safety check, run **\$INSTALL/bin/dbcheck mcddb** (UNIX) to verify the integrity of the database. Note, you must be user **root** to run **dbcheck**.
- No errors should be detected.
- 

## mcshadow Command Files

The **mcshadow** command uses the files listed in [Table 16-1](#).

**Table 16-1** *mcshadow Files*

File	Description
<b>mcddb.dbd</b>	Template file that describes the low-level data schema for the Raima runtime library.
<b>mcddb.k01</b> <b>mcddb.k02</b> <b>mcddb.k03</b>	Key files that contain the data that is redundant with the data files. Prime Access Registrar does not back up these files because they can be completely rebuilt with the <b>keybuild</b> command.
<b>mcddcd.d01</b> <b>mcddcd.d02</b> <b>mcddcd.d03</b>	Data files that contain the backup.
<b>mcdConfig.txt</b>	Text file from which Prime Access Registrar configures the initial at-install-time database.

**Table 16-1** *mcdshadow Files (continued)*

<b>File</b>	<b>Description</b>
<b>mcdschema.txt</b>	Text file that contains a version number denoting the level of the schema contained in the dbd file. Prime Access Registrar will not attempt to open the database unless the number in this file matches a constant that is hard-coded in the libraries. If the result of the mcdshadow command (which uses copies of the data files) is divorced from its original mcdschema.txt, you will not be able to run Prime Access Registrar.
<b>vista.taf</b> <b>vista.tcf</b> <b>vista.tjf</b>	Working files used by the Raima runtime library to ensure transactional integrity.





---

## Symbols

/bin/arserver [9-4](#)

---

## A

AcceptAll [2-28](#)

Access Registrar

definition [1-1](#)

internal database [16-1](#)

objects [1-2](#)

server [2-10](#)

Accounting [3-1](#)

attributes [1-6](#)

database [1-1](#)

definition [1-1](#)

log file [2-26](#)

MaxFileAge [3-3](#)

MaxFileAge format [3-4](#)

MaxFileSize [3-3](#)

MaxFileSize format [3-4](#)

RolloverSchedule [3-3](#)

setting up [3-2](#)

Start [3-1](#)

Stop [3-1](#)

Accounting records [8-11](#)

arbug [9-33](#)

arserver file [9-4](#)

Attribute Dictionary [1-6](#)

Attributes

check item [9-34](#)

AttributesToBeLogged [3-6](#)

AttributesToBeReturned [9-7](#)

Authorization [14-14](#)

definition [1-1](#)

---

## B

BackingStoreDiscThreshold [2-89, 2-133, 13-14](#)

BaseProfile [2-13](#)

blacklisting Diameter [4-24](#)

blacklisting SIGTRAN-M3UA [14-21](#)

---

## C

Callback-Number [1-6](#)

Change of Authorization (CoA) [9-40](#)

Check item attributes [9-34](#)

CIDR notation [2-120](#)

Cisco Prime Access Registrar

backups [16-1](#)

Classless Inter-Domain Routing [2-120](#)

Clients

IPAddress [2-120](#)

vendor properties [2-120](#)

CoA requests [9-40](#)

command authorization support [9-57](#)

Commands

eap-trace [5-45](#)

tunnel [5-45](#)

Configuration Examples

Query-Notify feature [9-8](#)

Configuring

check item attributes [9-35](#)

LDAP RemoteServer [12-3](#)

ODBC RemoteServer [13-7](#)

Configuring CoA requests [9-40](#)

Configuring rules [10-2](#)

CRB-Prepaid billing  
with SSG [8-15](#)

---

## D

Database

Cisco Prime Access Registrar backups [16-1](#)

DefaultAccountingService [2-11](#)

DefaultAuthenticationService [1-4, 2-10](#)

DefaultAuthorizationService [2-10](#)

DefaultSessionManager [2-10](#)

DefaultSessionService [2-11](#)

DetectOutOfOrderAccountingPacket [2-82](#)

diameter

session management [4-24](#)

diameter-eap [4-2](#)

Diameter Input queue

high threshold [15-5](#)

DNSLookupAndLDAPRebindInterval [12-6](#)

DropPacket. [2-28](#)

---

## E

EAP [5-1](#)

authentication mechanism [5-1](#)

EAP-AKA' [5-6](#)

eap-aka' [2-42](#)

eap-aka-prime (eap-aka') [5-7](#)

EAP authentication [5-1](#)

EAP-GTC [5-19, 5-20](#)

EAP-LEAP [5-21](#)

EAP-MD5 [5-22](#)

EAP-MSChapv2 [5-24](#)

EAP-Negotiate [5-23, 5-24](#)

EAP-SIM [5-26](#)

EAP-SIM authentication [5-28, 5-30](#)

eap-trace command [5-45](#)

EAP-Transport Level Security [5-31](#)

Easysoft Open Source [13-13](#)

EnableNotifications [2-121](#)

EnableRolloverIntelligence [3-5](#)

Environment Dictionary [1-4](#)

ExecCLIDRule [10-16](#)

ExecDNISRule [10-16](#)

ExecNASIPRule [10-17](#)

ExecRealmRule [10-15](#)

ExecTimeRule [10-12, 10-20](#)

Extensible Authentication Protocols [4-1, 5-1, 14-1](#)

---

## F

Failover policy [2-48](#)

fastrules [2-59, 11-1](#)

CLI [11-2](#)

GUI [2-59](#)

Fileaccounting

AttributesToBeLogged [3-6](#)

EnableRolloverIntelligence [3-5](#)

FileType [3-5](#)

file service [2-18, 2-26](#)

FileType [3-5](#)

Force update [9-12](#)

Framed-IP-Address [1-6](#)

Framed Protocol [1-6](#)

---

## G

Grouping property [10-2](#)

Group service [8-5, 8-13, 8-14](#)

GUI

launching [2-1](#)

logging in [2-3](#)

log out [2-4](#)

---

**H**

Hot configuration [6-6](#)  
 Hot-lining [9-40](#)

---

**I**

IncomingScript [2-10, 2-66, 2-117](#)  
 Incoming scripts [1-3](#)  
 Information collection  
     automatic [9-33](#)  
 InitEntryPoint [2-19](#)  
 Input queue  
     high threshold [15-5](#)  
 internal scripts [2-18](#)  
     GUI [2-19](#)  
 IPAddress [2-120](#)

---

**L**

LDAP [12-1](#)  
     hostname [12-3](#)  
     MultipleServersPolicy [12-2](#)  
 LDAP Accounting [2-127](#)  
 LDAP Rebind [12-6](#)  
     failures [12-6](#)  
 LDAP RemoteServer [12-3](#)  
 LDAP service [12-2](#)  
 LDAPToCheckItemMappings [12-7](#)  
 LDAPToEnvironmentMappings [12-7](#)  
 LDAPToRadiusMappings [12-7](#)  
 LEAP [5-21](#)  
 Lightweight Directory Access Protocol [12-1](#)  
 local [2-27](#)  
     UserList type [2-30](#)  
 local service [2-30](#)  
 Log files  
     file system [3-4](#)  
     managing [3-3](#)

Logging in  
     GUI [2-3](#)  
 Login page [2-3](#)

---

**M**

m3ua service  
     map restore data [14-14](#)  
 map restore data [14-14](#)  
     authorization flow [14-14](#)  
     CLI [14-16](#)  
     insert subscriber data [14-15](#)  
 map restore data authorization [14-14](#)  
     CLI configuration [14-16](#)  
     flow [14-14](#)  
     insert subscriber data structure [14-15](#)  
 Master-URL-Fragment [9-11](#)  
 MCD [16-1](#)  
 mcdcd.d01-d03 [16-2](#)  
 mcdConfig.txt [16-2](#)  
 mcddb.dbd [16-2](#)  
 mcddb.k01-k03 [16-2](#)  
 mcdshadow [16-1](#)  
 Measurements  
     prepaid billing [8-7](#)  
 Microsoft WPS [9-9](#)  
 multiple [1-1](#)  
 MultipleServersPolicy [12-2, 13-7](#)

---

**N**

NAS [1-1, 3-1](#)  
 NAS-IP-Address [1-6](#)  
 NAS-Port [1-6](#)  
 NetMask [2-121](#)  
 NotificationProperties [2-122](#)

**O**

ODBC.ini file [13-2](#)  
 ODBCDataSource [13-10, 13-13](#)  
 ODBC RemoteServer [13-7](#)  
 ODBC service [13-6](#)  
 ODBCToEnvironmentMappings [13-12](#)  
 ODBCToRadiusMappings [13-12](#)  
 ORACLE\_HOME [13-2](#)  
 Oracle Driver  
     Easysoft Open Source [13-13](#)  
 Oracle functions [13-11](#)  
 OS paging size [2-104](#)  
 OutagePolicy [2-28](#)  
 OutageScript [2-28](#)  
 OutgoingScript [2-10, 2-66, 2-117](#)  
 Outgoing scripts [1-3](#)  
 Overview [1-1](#)

**P**

Packet buffering [3-11](#)  
 Packet fields [1-6](#)  
 Packet of disconnect [9-36](#)  
 Paging size (operating system) [2-104](#)  
 ParseTranslationGroupsByCLID [10-10, 10-21](#)  
 ParseTranslationGroupsByDNIS [10-10, 10-20, 10-21](#)  
 ParseTranslationGroupsByReal [10-20](#)  
 ParseTranslationGroupsByRealm [10-10](#)  
 Password  
     length of [2-16](#)  
 Password change [9-12](#)  
 PCO-Parse-Client-Outgoing [8-15](#)  
 PEAP Version 0 [5-46](#)  
 PEAP Version 1 [5-51](#)  
 PhantomSessionTimeout [2-106](#)  
 Policies  
     configuring [10-1](#)  
     validation [10-3](#)

Policy [10-1](#)  
 Policy engine  
     attribute translation [10-9](#)  
     parsing translation groups [10-10](#)  
     reducing overhead [10-13](#)  
     time of day access restrictions [10-11](#)  
     wildcard support [10-2](#)  
 Port 8080 [2-1](#)  
 PPO-Parse-Prepaid-Outgoing [8-16](#)  
 PPP [1-3, 1-6, 2-11](#)  
 Prepaid  
     AA service [8-5, 8-12](#)  
     group service [8-5, 8-13, 8-14](#)  
 Prepaid billing  
     measurements [8-7](#)  
 Protected EAP [5-1](#)

**Q**

Query-Notify [9-6](#)  
 Query-Notify AttributeGroup  
     configuration example [9-9](#)  
 Query-Notify client  
     configuration example [9-9](#)  
 quintets-triplets conversion [5-30](#)

**R**

radclient  
     testing EAP-TTLS [5-42](#)  
 radclient commands [5-44](#)  
 RADIUS  
     messages [1-5](#)  
 RADIUS\_WORKER\_THREAD\_COUNT [9-4](#)  
 RADIUS packet fields [1-6](#)  
 RadiusServer object [1-2](#)  
 radius to diameter [4-21](#)  
 RejectAll [2-28](#)

- RemoteServers [12-2, 13-7](#)
  - Renewal [9-12](#)
  - RepIPMaster [6-9](#)
  - Replication
    - archive [6-4](#)
    - automatic resynchronization [6-5](#)
    - configuration settings [6-6](#)
    - data flow [6-3](#)
    - data integrity [6-4](#)
    - hot configuration [6-6](#)
    - hot-standby [6-1](#)
    - impact on request processing [6-6](#)
    - RepIPAddress [6-8](#)
    - RepTransactionArchiveLimit [6-2, 6-8](#)
    - RepTransactionSyncInterval [6-2, 6-7, 6-10, 6-11](#)
    - security [6-4](#)
    - slaves [6-9](#)
    - slave server [6-3](#)
    - transaction order [6-5](#)
    - transaction verification [6-4](#)
  - RepMasterIPAddress [6-9](#)
  - RepMasterPort [6-9](#)
  - RepPort [6-8](#)
  - RepSecret [6-8](#)
  - RepType [6-7](#)
  - Resource allocation
    - dynamic [1-4](#)
  - resource manager
    - 3gpp [1-5, 2-108](#)
    - dynamic-dns [1-4, 2-108](#)
    - group-session-limit [1-4, 2-108](#)
    - home-agent [1-4, 2-108](#)
    - home-agent-IPv6 [1-4, 2-108](#)
    - ip-dynamic [1-4, 2-108](#)
    - ip-per-NAS-port [1-4, 2-108](#)
    - ipx-dynamic [1-4, 2-108](#)
    - remote-group-session-limit [1-4, 2-108](#)
    - remote-ip-dynamic [1-4, 2-108](#)
    - remote-session-cache [1-5, 2-108](#)
    - remote-user-session-limit [1-4, 2-108](#)
    - session-cache [1-4, 2-108](#)
    - subnet-dynamic [1-4, 2-108](#)
    - user-session-limit [1-4, 2-108](#)
    - usr-vpn [1-4, 2-108](#)
  - Resource Managers [1-5](#)
  - Resynchronization
    - automatic [6-5](#)
    - full [6-5](#)
  - REX
    - scripts [2-18](#)
  - RFC
    - 2866 [3-1](#)
  - RolloverSchedule [3-3](#)
    - time format [3-5](#)
  - RoundRobin policy [2-48](#)
  - Routing requests [10-4](#)
    - based on CLID [10-6](#)
    - based on DNIS [10-5](#)
    - based on NASIP [10-7](#)
    - based on realm [10-4](#)
    - based on User-Name Prefix [10-8](#)
  - Rules
    - script and attribute requirements [10-3](#)
    - standard scripts [10-15](#)
- 
- ## S
- Scripts
    - types of [1-3](#)
  - SCTP multihoming [14-21](#)
  - SelectPolicy [10-1](#)
  - Send-PEAP-URI-TLV [9-11](#)
  - Server
    - master [6-1](#)
    - primary [6-1](#)
    - secondary [6-1](#)
  - Services
    - file [2-26](#)

local [2-27, 2-30](#)  
 used for [1-3](#)  
 services [2-25](#)  
 Services objects [2-25](#)  
 Session magic number [2-82](#)  
 Session Management  
     definition [1-1](#)  
     types of [1-4](#)  
 Session record size [2-104](#)  
 Shadow backups [16-1](#)  
 Shared secret [2-120](#)  
 Sign up [9-11](#)  
 Sign-up URL [9-10](#)  
 sigtran-m3ua  
     SCTP multihoming [14-21](#)  
 SLIP [1-6](#)  
 SNMP [15-1](#)  
     Trap recipients [15-12](#)  
     traps [15-3](#)  
 SNMP Configuration  
     community string [15-12](#)  
 SQLDefinition [13-10](#)  
 SQL queries [13-11](#)  
 SQLStatement [13-10](#)  
 SQL syntax restrictions [13-11](#)

---

## T

TACACS+  
     command authorization [9-57](#)  
     command authorization flow [9-59](#)  
 translation  
     diameter to radius [4-21](#)  
 Trap recipients [15-12](#)  
 Traps  
     carAccountingLoggingFailure [15-7](#)  
     carInputQueueFull [15-5](#)  
     carInputQueueNotVeryFull [15-5](#)  
     carOtherAccServerResponding [15-7](#)

    carOtherAuthServeNotrResponding [15-7](#)  
     carOtherAuthServerResponding [15-6](#)  
     carServerStart [15-5](#)  
     carServerStop [15-5](#)  
     configuring [15-12](#)  
     supported [15-4](#)  
 tunnel command [5-45](#)

---

## U

UNIX directories [1-2](#)  
 UserGroups  
     check item attributes [9-35](#)  
 UserList [1-2](#)  
     check item attributes [9-35](#)  
 User objects [1-3](#)  
 User profiles [1-3](#)  
 UserService [5-22](#)

---

## W

WAP [9-6](#)  
 Windows 95 Registry [1-2](#)  
 Windows Provisioning Service (WPS) [9-9](#)  
 Wireless Application Protocol [9-6](#)

---

## X

XML Query Identity [7-2](#)