



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. 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. 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 /Radius/Services/ODBC/RemoteServers such as ODBC-Primary and ODBC-Secondary

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); UserPasswordAttribute = (mandatory, no default; data store field for user password) SQLStatements/ SQLStatement1/ SQLStatement2/
ODBCToRadiusMappings (OCIToRadiusMappings)	Optional; a list of name/value pairs in which the name is the name of the odbc attribute to retrieve from the user record, and the value is the name of the RADIUS attribute to set to the value of the odbc attribute retrieved. For example, when the ODBCToRadiusMappings has the entry: FramedIPAddress = Framed-IP-Address , the RemoteServer retrieves the FramedIPAddress attribute from the odbc user entry for the specified user, uses the value returned, and sets the Response variable Framed-IP-Address to that value.
	 Note When you select the protocol as OCI, the field name will be displayed as OCIToRadiusMappings.

Table 13-2 ODBC/OCI Remote Server Parameters (continued)

Parameter	Description
ODBCToEnvironmentMappings (OCIToEnvironmentMappings)	<p>Optional; a list of name/value pairs in which the name is the name of the odbc 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>For example, when the ODBCToEnvironmentMappings has the entry: group = User-Group, the RemoteServer retrieves the group attribute from the odbc user entry for the specified user, uses the value returned, and sets the Environment variable User-Group to that value.</p> <p> Note When you select the protocol as OCI, the field name will be displayed as OCIToEnvironmentMappings.</p>
ODBCToCheckItemMappings (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 ODBCToCheckItemMappings has the entry: group = User-Group, the Access Request must contain the attribute group, and it must be set to User-Group.</p> <p> Note 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

```

