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

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

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

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

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 odbc-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 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 **/Radius/Advanced/ODBCDataSources**. 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

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

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

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 **BindName**.
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 insetting 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 insetting 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 **MaxOutstandingRequests** 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 **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.
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 **CertificateDBPath** field in the **Advanced** 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 ‘AttribtuesToWrite’ 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 ]
    seealso = 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 **AcceptAll**, **DropPacket**, or **RejectAll**. Default is **DropPacket.**
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 /Radius/Services/LDAP/RemoteServers. 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 LADP-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 **BindName**.
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 insetting 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 insetting for information about user joe, use the fentry 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 **MaxOutstandingRequests** 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 **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.
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 **CertificateDBPath** field in the **Advanced** 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

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