## Contents

**Preface** .................................................................................................................................................................... vii  
   Audience ................................................................................................................................................................... vii  
   Downloading Oracle Communications Documentation .............................................................. vii  
   Documentation Accessibility ......................................................................................................................... vii  
   Document Revision History ............................................................................................................................... vii  

1 **Using BRM AAA Gateway Manager**

   About AAA Gateway Manager....................................................................................................................... 1-1  
   About the AAA Gateway Manager System Architecture ........................................................................... 1-2  
      Dispatcher Pipeline ................................................................................................................................. 1-3  
      Processing Pipeline ............................................................................................................................... 1-4  
      Output Pipeline .................................................................................................................................... 1-4  
      Timeout Pipeline ................................................................................................................................... 1-4  
      Exception Pipeline ................................................................................................................................. 1-5  
      Replay Pipeline ................................................................................................................................... 1-5  
      MBI_CDR_PlaybackPipeline ..................................................................................................................... 1-6  
      Communication Between Pipelines...................................................................................................... 1-6  
   About Supported Formats for Network Requests ....................................................................................... 1-6  
   About Installing AAA Gateway Manager ................................................................................................. 1-7  
      Installing AAA Gateway Manager ...................................................................................................... 1-7  
      Uninstalling AAA Gateway Manager ................................................................................................... 1-9  
      Installing HP OpenCall Manager .......................................................................................................... 1-9  
      Uninstalling HP OpenCall Manager ...................................................................................................... 1-9  
      Installing Diameter Charging Manager .............................................................................................. 1-9  
      Uninstalling Diameter Charging Manager .......................................................................................... 1-10  
   About Configuring AAA Gateway Manager ............................................................................................. 1-10  
      Configuring AAA Gateway Manager ................................................................................................... 1-11  
   Starting AAA Gateway Manager ............................................................................................................... 1-12  
      Stopping AAA Gateway Manager ........................................................................................................ 1-13  
   About Diameter Transport Failure Detection .............................................................................................. 1-13  
   Specifying the Maximum Number of CM Connections for AAA Gateway ............................................... 1-14  
   Rebalancing CM Connections ................................................................................................................... 1-14  
   Troubleshooting AAA Gateway Manager .................................................................................................. 1-15  
      Enabling OMF Instrumentation Data Collection in AAA Gateway Manager .................................... 1-15  
      Getting Diagnostic Data from AAA Gateway Manager ....................................................................... 1-15
2 AAA Gateway Manager Protocol Support

About AAA Gateway Manager Protocol Support ................................................................. 2-1

Diameter Protocol Support Files ............................................................................................ 2-2
  Diameter Input Grammar ....................................................................................................... 2-3
  Diameter Field Blocks ............................................................................................................ 2-3
  Error Handling ....................................................................................................................... 2-4
  Diameter Input Mapping ....................................................................................................... 2-4
  Diameter Stream Format Description .................................................................................... 2-5
  Diameter Output Grammar .................................................................................................... 2-6
  Error Handling ....................................................................................................................... 2-6
  Diameter Output Mapping ..................................................................................................... 2-6
  Supported Diameter Messages ............................................................................................. 2-6
  Diameter Error Messages ..................................................................................................... 2-22

MBI Protocol Support Files ................................................................................................... 2-23
  MBI Input Grammar .............................................................................................................. 2-23
  Error Handling ...................................................................................................................... 2-23
  MBI Input Mapping ............................................................................................................... 2-24
  MBI Stream Format Description .......................................................................................... 2-24
  MBI Output Grammar .......................................................................................................... 2-24
  MBI Output Mapping ............................................................................................................ 2-24
  Error Handling ...................................................................................................................... 2-24
  MBI Service Types Supported and Their Mapping to BRM Objects ..................................... 2-24
  Supported MBI Messages ..................................................................................................... 2-25
  About Supporting Roaming Requests .................................................................................... 2-27
  About Supporting Call-Forwarding Requests ....................................................................... 2-27

MBI CDR Protocol Support Files .......................................................................................... 2-28
  MBI CDR Input Grammar .................................................................................................... 2-28
  Error Handling ...................................................................................................................... 2-28
  MBI CDR Input Mapping ...................................................................................................... 2-28
  MBI CDR Stream Format Description .................................................................................. 2-28

Customizing AAA Gateway Manager Protocol Support ...................................................... 2-29
  Example of Configuring the Gateway to Support Additional MBI Data ............................. 2-29
3 AAA EDR Container Description

About the AAA EDR Container Description ................................. 3-1
HEADER Content Type for AAA EDRs.............................................. 3-1
DETAIL Content Type for AAA EDRs .............................................. 3-2
  Function Module Fields ................................................................. 3-2
  ASSOCIATED_SCP_REQUEST Block ............................................. 3-5
  ASSOCIATED_OPERATIONAL_REQUEST Block ........................... 3-5
  ASSOCIATED_TCF_AAA_DETAIL block ....................................... 3-5
  PCM_OP_TCF_AAA_AUTHORIZE Block ......................................... 3-6
  PCM_OP_TCF_AAA_AUTHORIZE Block ......................................... 3-7
  PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE Block ............. 3-7
  PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION Block ..................... 3-7
  PCM_OP_TCF_AAA_START_ACCOUNTING Block ......................... 3-8
  PCM_OP_TCF_AAA_UPDATE_ACCOUNTING Block ....................... 3-8
  PCM_OP_TCF_AAA_STOP_ACCOUNTING Block ............................. 3-8
  PCM_OP_TCF_AAA_QUERY_BALANCE Block .................................. 3-9
  PCM_OP_TCF_AAA_SERVICE_PRICE_ENQUIRY Block .................. 3-9
  PCM_OP_TCF_AAA_ACCOUNTING_ON Block ................................ 3-9
  PCM_OP_TCF_AAA_ACCOUNTING_OFF Block ................................ 3-10
  PCM_OP_TCF_AAA_REFUND Block .............................................. 3-10
  ASSOCIATED_PROTOCOL_INFO Block ....................................... 3-10
  PCM_OP_EBUF Block ................................................................. 3-11
INTERNAL Record in AAA EDRs .................................................... 3-11
Customizing the AAA EDR Container Description ....................... 3-13
Preface

This book provides an overview of the Oracle Communications Billing and Revenue Management (BRM) AAA Gateway Manager. It also provides an overview of the network message protocols supported by AAA Gateway Manager.

Audience

This document is intended for developers and system administrators.

Downloading Oracle Communications Documentation

Product documentation is located on Oracle Technology Network:

http://docs.oracle.com

Additional Oracle Communications documentation is available from the Oracle software delivery Web site:

http://edelivery.oracle.com

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Document Revision History

The following table lists the revision history for this book.

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E25078-01</td>
<td>March 2013</td>
<td>Initial release.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Made minor formatting and text changes.</td>
</tr>
</tbody>
</table>
Using BRM AAA Gateway Manager

This document provides an overview of Oracle Communications Billing and Revenue Management (BRM) AAA Gateway Manager, which receives and responds to authentication, authorization, and accounting (AAA) requests from the network for prepaid services. In addition, this document describes how to:

- Install and configure AAA Gateway Manager.
  See "About Installing AAA Gateway Manager" for more information.
- Install and configure the optional HP OpenCall Manager and Diameter Charging Manager.
  See "Installing HP OpenCall Manager" and "Installing Diameter Charging Manager" for more information.
- Monitor and maintain AAA Gateway Manager.
  See "Rebalancing CM Connections" for more information.

---

**Important:** AAA Gateway Manager is an optional feature that requires a separate license.

---

### About AAA Gateway Manager

AAA Gateway Manager is a BRM component that interfaces between BRM and the network Service Control Point (SCP). It performs the following functions:

- Translates AAA requests that use any network protocol into Portal Communication Protocol (PCP) required by BRM.
- Contains the business logic for processing AAA requests and provides a way for you to configure custom business logic solutions.
- Provides a failover mechanism if an RCP component fails, by connecting to the next available CM when a CM becomes unavailable.
- Processes requests from the network in a degraded mode until the failed RCP component is fully functional.
- Distributes the requests from the network among the CMs in the connection pool and balances loads.
- Maps the asynchronous requests from the network to synchronous requests in CM by using queues between the pipelines.
AAA Gateway Manager is a stateless component that you can configure for high availability by allowing the network to switch to a standby AAA Gateway Manager if the primary one fails.

**Important:** The AAA Gateway Manager processes AAA requests and then calls BRM AAA opcodes only. Other requests that are routed to non-AAA opcodes should be sent through your CRM application rather than the AAA Gateway Manager.

### About the AAA Gateway Manager System Architecture

AAA Gateway Manager is a set of real-time pipelines run by the BRM Pipeline Manager. AAA Gateway Manager consists of the following set of pipelines, each of which runs in its own thread and has its own input buffer:

- Dispatcher Pipeline
- Processing Pipeline
- Output Pipeline
- Timeout Pipeline
- Exception Pipeline
- Replay Pipeline
- (Optional with HP OpenCall Manager) MBI_CDR_PlaybackPipeline

You can configure multiple pipelines to process requests.

*Figure 1–1* shows a high-level view of a *basic* configuration for AAA Gateway Manager:
Dispatcher Pipeline

The **Dispatcher pipeline** reads the data from the network and translates it into an EDR. See "About Supported Formats for Network Requests" for more information.

Depending on the type of request, the FCT_IRules routing module in the pipeline passes the request to the appropriate Processing pipeline based on the service and request type. For example, a GSM service authorization request is passed to the processing pipeline that has the business logic to process GSM authorization requests.

The default router, **IRL_ROUTER.irl** in the Pipeline_home/iScriptLib/AAA directory, sets timeout values based on the type of request and routes the requests to the appropriate pipeline. You can customize this module or replace it with your own custom module.


The Dispatcher pipeline includes FCT_Timer, which sets the timeout limit for each request and starts the timer. FCT_Timer stores a copy of the original EDR before sending the EDR to the Processing pipeline. If the EDR is processed successfully, the timer is stopped. When an EDR times out, the Dispatcher pipeline sends the copy of the EDR to the Timeout pipeline. The original EDR is marked as timed out and is sent to the Exception pipeline if any further processing is necessary.

The default timeout limit is 200 microseconds. You can change it to suit your requirements by editing the timeout entry in the registry file. You can specify a different timeout value for different types of requests, such as authorization, start accounting, and stop accounting.
See "FCT_Timer" in BRM Configuring Pipeline Rating and Discounting for more information on FCT_Timer.

Processing Pipeline

The Processing pipeline contains a sequence of modules to convert the EDR to an input flist, call BRM opcodes, and convert the results returned by the opcode to an EDR.

The FCT_NumberPortability module gets number portability data from the DAT_NumberPortability module and determines the network operator that applies.

The ISC_PopulateOpcodeandUtilBlock_Diameter.isc iScript which you can customize, adds an opcode block in the EDR.

The ISC_MapNetworkOperatorInfo module maps the source, origin, and destination network fields to the destination network field of the opcode input block.

The FCT_Opcode module in the pipeline uses the DAT_ConnectionPool module to connect to the Connection Manager (CM) and call the appropriate opcode for the request.

The post-opcode module, ISC_GenericPostOpcode.isc, which is an iScript that you can customize, checks the results of the opcode call for ebuf and timer state. The module then sends the EDR to the Exception pipeline if the response timed out because of any BRM component failure.

The Processing pipeline also includes FCT_CancelTimer, which cancels the timer if the EDR is successfully processed within the time limit.

See "FCT_CancelTimer" in BRM Configuring Pipeline Rating and Discounting for more information.

By default, each processing pipeline can process authorization, start accounting, and stop accounting requests.

Output Pipeline

The Output pipeline translates the EDR containing the AAA results back to the protocol-specific representation and sends it to Intelligent Network (IN). The Output pipeline collects all the results from the pipelines, makes a single connection, and sends them to the network.

See "About Supported Formats for Network Requests" for information about network protocols supported.

Timeout Pipeline

The Timeout pipeline implements a degraded process mode for timed-out EDRs from any of the processes and authorizes or rejects a request based on a policy that you specify. The Timeout pipeline includes the following pipelines:

- **TimeoutRouter pipeline** - This pipeline uses the rules in the IRL_TimeoutRouter.irl module to direct the authorization and reauthorization requests to the TimeoutProcess pipeline and other requests to the TimeoutDefault pipeline. You can customize the IRL_TimeoutRouter.irl module in the Pipeline_home/iScript/AAA directory to implement your own routing rules.

- **TimeoutDefault pipeline** - This pipeline uses the ISC_TimeoutDefaultResponse.isc iScript to send a standard response declining all requests except authorization and reauthorization. You can customize ISC_
TimeoutDefaultResponse.isc in the Pipeline_home/iScript/AAA directory to change the default response or to implement custom business logic.

- **TimeoutProcess pipeline** - This pipeline processes the authorization and reauthorization requests and sends a response to the network.

By default, a set of iScript modules in this pipeline identifies the category of the call (such as free call or emergency call) and call zones (such as local, long distance, or international) based on a static set of rules and using the impact category. This allows the module to authorizes free, local, and emergency calls and reject other requests.

The following iScript modules are also included in this pipeline and you can customize them and implement your own business logic to handle the timed-out requests:

- **ISC_TimeoutPreZone.isc** - Populates the fields in the request required for the zone module.
- **ISC_TimeoutProcessedResponse.isc** - Based on a static set of configurable rules and the impact category determined by the FCT_Zone module, this module authorizes or rejects a request.

The AAA Gateway Manager framework marks an EDR as timed out if it is not processed because of a BRM process failure within the time interval that you configure. The original EDR is marked as timed out and sent to the Exception pipeline for playback in the future when a failed process starts again.

**Exception Pipeline**

The *Exception pipeline* records EDRs from any of the pipelines that have been marked as containing an error by the business logic components. The Exception pipeline also records timed-out EDRs. By default, the Exception pipeline stores requests such as stop accounting for later replay. You can configure it to store any EDRs that you want to replay; for example, EDRs from the stop accounting process.

To prevent revenue loss, the saved EDRs are written by default to an XML file and replayed later by the Replay pipeline.

**Replay Pipeline**

The *Replay pipeline* processes timed-out EDRs when BRM becomes available. For example, if a timed-out call request was authorized or if an activity was recorded in the BRM database when the timeout occurred, you can capture those activities and record them in the database.

By default, the Replay pipeline includes the following modules that you can customize:

- **IRL_ReplayRouter.irl** - Enables replaying of only end call, IN startup, and IN shutdown requests.
- **ISC_ReplayPostOpcode.isc** - When replaying the serialized requests, if there are any errors, logs the errors. If the RCP is down again, it sends the requests to the Exception pipeline for replaying later.

---

**Note:** These modules are installed in the Pipeline_home/iScriptlib/AAA directory.
You can implement your business logic by customizing an iScript to process the EDRs and record the results in BRM database, if necessary.

**MBI_CDR_PlaybackPipeline**

---

**Important:** This pipeline is installed only when you install HP OpenCall Manager to process requests that use the Message Based Interface (MBI) protocol.

---

The *MBI_CDR_PlaybackPipeline* pipeline processes requests in CDR format. When the AAA requests a timeout because the prepaid system becomes unavailable, the Open Call Service Access Controller (OCSAC) records the END requests as CDRs in files. When the prepaid system becomes available, you can send the CDRs to AAA Gateway Manager for processing.

The MBI_CDR_PlaybackPipeline pipeline is similar to the Replay pipeline. It contains a sequence of modules to convert the CDR to an input flist and to call BRM opcodes that process the request, calculate any charges, and update the balances.

The FCT_Opcode module in this pipeline interacts with CM and calls the appropriate opcode for the request.

See "FCT_Opcode" in *BRM Configuring Pipeline Rating and Discounting* for more information.

**Communication Between Pipelines**

AAA Gateway Manager uses queues to communicate between the pipelines. The queues are created at startup before the pipelines are instantiated and are used by pipelines to order the requests for processing. Each pipeline; for example, the Processing pipeline or the Timeout pipeline have their own dedicated input queues, from which the pipeline’s input module reads the EDR container. The output module in each pipeline writes the EDR container to a queue for the next pipeline.

---

**Note:** The output from a pipeline can also be written to a file or a database, depending on the pipeline. For example, by default, the output of the Exception pipeline is written to an XML file.

---

**About Supported Formats for Network Requests**

AAA Gateway Manager can process data only in EDR container format. InSocketManager in the Dispatcher pipeline reads requests from IN, which can be in HP OpenCall MBI or Diameter protocol formats, and constructs an EDR container that a pipeline can process. The EXT_OutSocketManager module in the Output pipeline converts the responses to the requests, which are EDR containers, to the format required by the IN and sends them to the IN over TCP/IP.

---

**Note:** When the Dispatcher pipeline receives MBI requests in CDRs, it forwards the CDRs to the MBI_CDR_PlaybackPipeline pipeline for processing.
About Installing AAA Gateway Manager

To install and use AAA Gateway Manager, you must install and configure the following software:

1. BRM.
2. Third-Party software, which includes the Perl libraries and JRE required for installing BRM components.
3. AAA Gateway Manager.
   See "Installing AAA Gateway Manager" for more information.
4. (Optional) BRM HP OpenCall Manager, to support network requests using MBI protocol.
   See "Installing HP OpenCall Manager" for more information.
5. (Optional) BRM Diameter Charging Manager, to support network requests using Diameter protocol.
   See "Installing Diameter Charging Manager" for more information.

Installing AAA Gateway Manager

1. Download the software to a temporary directory (temp_dir).
About Installing AAA Gateway Manager

---

**Important:** You must increase the heap size used by the Java Virtual Machine (JVM) before running the installation program to avoid "Out of Memory" error messages in the log file. See "Increasing Heap Size to Avoid “Out of Memory” Error Messages” in BRM Installation Guide or more information.

---

2. To set up environment variables for JRE and Perl libraries for installing AAA Gateway Manager, go to the **ThirdParty** installation directory and source the **source.me.sh** for the shell:

   ```
   % source source.me.sh
   ```

---

**Note:**

- The default installation directory for the Third-Party software is `opt/portal/ThirdParty`.
- The **source.me.sh** is for a bash shell. If you use a C shell, enter **source.me.csh**.

---

3. Go to **temp_dir** and enter this command to install AAA Gateway Manager:

   ```
   7.4_AAA_Gateway_platform_64_opt.bin
   ```

**Note:** You can use the `-console` parameter to run the installation in a command-line mode. To use the graphical user interface (GUI) for installation, make sure you have X Windows installed in your computer.

---

4. (Optional) If you want to install AAA Gateway Manager server components separately, either on this computer or on another computer, select custom install when asked to specify the setup type.

   Select the components you are installing by typing their respective numbers and click **Next**.

   The components are:

   - **AAA Gateway Files**
   - **Framework**

5. When prompted, enter the directory where you want to install AAA Gateway Manager.

   The default directory for installing AAA Gateway Manager is **opt/ifw**.

**Important:** If you are also using Pipeline Manager, make sure you install AAA Gateway Manager in the same directory as Pipeline Manager.

---

The installation program installs AAA Gateway Manager and the Pipeline Manager framework files in the specified directory.
Uninstalling AAA Gateway Manager
To uninstall AAA Gateway Manager, run the Pipeline_home/uninstaller/AAA_Gateway/uninstaller.bin.

Installing HP OpenCall Manager
1. Download the software to a temporary directory (temp_dir).

Important: You must increase the heap size used by the Java Virtual Machine (JVM) before running the installation program to avoid "Out of Memory" error messages in the log file. See "Increasing Heap Size to Avoid “Out of Memory” Error Messages “ in BRM Installation Guide for more information.

2. To set up environment variables for JRE and Perl libraries for installing HP OpenCall Manager, go to the ThirdParty installation directory and source the source.me.sh for the shell:
   % source source.me.sh

Note:
- The default installation directory for the ThirdParty software is opt/portal/ThirdParty.
- The source.me.sh is for a bash shell. If you use a C shell, enter source.me.csh.

3. Go to temp_dir and enter this command to install HP OpenCall Manager:
   7.4_HPOpencall_platform_64_opt.bin

Note: You can use the -console parameter to run the installation in a command-line mode. To use the graphical user interface (GUI) for installation, make sure you have X Windows installed on your computer.

HP OpenCall Manager is installed in same directory where AAA Gateway Manager is installed.

Uninstalling HP OpenCall Manager
To uninstall HP OpenCall Manager, run the Pipeline_home/uninstaller/HPOpenCall/uninstaller.bin.

Installing Diameter Charging Manager
1. Download the software to a temporary directory (temp_dir).
Important: You must increase the heap size used by the Java Virtual Machine (JVM) before running the installation program to avoid “Out of Memory” error messages in the log file. See “Increasing Heap Size to Avoid “Out of Memory” Error Messages” in BRM Installation Guide for more information.

2. To set up environment variables for JRE and Perl libraries for installing Diameter Charging Manager, go to the Third_Party installation directory, and source the `source.me.sh` for the shell:

```
% source source.me.sh
```

Note:

- The default installation directory for the Third-Party software is `opt/portal/ThirdParty`.
- The `source.me.sh` is for a Bash shell. If you use a C shell, enter `source.me.csh`.

3. Go to `temp_dir` and enter this command to install the Diameter Charging Manager:

```
7.4_Diameter_platform_64_opt.bin
```

Note: You can use the `-console` parameter to run the installation in command-line mode. To use the graphical user interface (GUI) for installation, make sure you have X Windows installed in your computer.

Diameter Charging Manager is installed in the same directory where AAA Gateway Manager is installed.

Uninstalling Diameter Charging Manager

To uninstall Diameter Charging Manager, run the `Pipeline_home/uninstaller/Diameter/uninstaller.bin`.

About Configuring AAA Gateway Manager

AAA Gateway Manager includes a registry file with default entries that you can modify.

To configure AAA Gateway Manager, you need to perform the following tasks:

- Configure the Gateway to connect to the CM or CM pool in a high-availability system.
  
  For information on high-availability BRM systems, see BRM System Administrator's Guide.

- Configure the pipelines with any custom modules you have implemented.

- Modify the default entries.
Configuring AAA Gateway Manager

When you install AAA Gateway Manager, a registry file with default values for the entries is installed in the directory where you installed AAA Gateway Manager.

---

**Note:** The MBI.reg file is installed with HP OpenCall Manager; and the diameter_charge.reg file is installed with Diameter Charging Manager.

---

You can change the configuration to suit your installation.

1. Open the *.reg file from the opt/ifw/ directory.
2. Specify the format of your network requests, such as MBI, and the grammar mapping files for input and output mapping in the **DataDescription** entry.
3. Specify the path to the EDR container.
4. In the **InputStream** section, configure one of the following modules:
   - EXT_InSocketMgrDiameter
   - EXT_InSocketMgrFlist
   - EXT_InSocketMgrMbi
   See **BRM Configuring Pipeline Rating and Discounting** for more information on these modules.

Specify the port number for AAA Gateway Manager to listen to MBI or Diameter requests.
For example:
```
InputStream
{
    ModuleName = EXT_InSocketMgrFlist
    Module
    {
        Port = 11960
    }
}
```

5. Configure the following modules in the function pool sections of the pipelines:
   - FCT_Timer in the Dispatcher pipeline block. See "FCT_Timer" in **BRM Configuring Pipeline Rating and Discounting** for more information.
   - FCT_Opcode in the Processing pipeline block. See "FCT_Opcode" in **BRM Configuring Pipeline Rating and Discounting** for more information.
   - FCT_CancelTimer in the Processing pipeline block. See "FCT_CancelTimer" in **BRM Configuring Pipeline Rating and Discounting** for more information.
   - The custom module that you implement in the Replay pipeline.

6. Specify the path to any custom iScript or iRules modules you create in the pipelines.

7. (Optional) Change the size of the queues and buffers.

8. (Optional) If the Diameter credit control application in the external network uses accounting commands, add this entry to the ifw section of the diameter_charge.reg file:
Starting AAA Gateway Manager

AAA Gateway Manager requires access to the BRM Data Dictionary to start up. AAA Gateway Manager must start up once successfully before it can run in degraded mode.

---

**Note:** To run AAA Gateway Manager in degraded mode before a successful startup, you can use `testnap` and execute the PCM_OP_GET_DD opcode. The returned flist can be stored in a file and that file can be configured in the registry as the InfranetDataDictionaryFileName.

---

1. To set the environment variables to run AAA Gateway Manager, go to the AAA Gateway Manager installation directory and source the `source.me.sh` for the shell:

   ```bash
   % source source.me.sh
   ```

   **Note:**
   - The default installation directory for AAA Gateway Manager is `opt/ifw`.
   - The `source.me.sh` is for a bash shell. If you use a C shell, enter `source.me.csh`.

2. Use the following command to start AAA Gateway Manager:

   ```bash
   ifw -r RegistryFile | -h | -v [-r RegistryFile]
   ```

   where:
   - `-r RegistryFile`
   Starts the AAA Gateway pipelines with the specified registry file.
   - `-h`
   Displays the syntax and parameters.
   - `-v [-r RegistryFile]`
   Displays the version of the pipeline framework. If you use the `-r` parameter, it also displays the version and name of data and function modules. For example:

   ```bash
   ifw -v -r conf/diameter_charge.reg
   ```

   AAA Gateway Manager displays **Starting to process** when startup procedures have completed.
Stopping AAA Gateway Manager

You stop AAA Gateway Manager by using the following semaphore entry:

\[
\text{ifw.Active} = \text{False}
\]

AAA Gateway Manager shuts down after 10 or more seconds.

See "Updating Configuration Settings During Runtime by Using Semaphore Files in BRM System Administrator’s Guide for more information on semaphores and how to create semaphore files.

About Diameter Transport Failure Detection

AAA Gateway Manager detects transport failures between the Diameter IN and itself by sending Device Watchdog Answer (DWA) and Device Watchdog Request (DWR) messages to the Diameter IN. Transport failure detection helps prevent zombie socket connections between the Diameter client and AAA Gateway Manager. To detect transport failures, AAA Gateway Manager follows these steps:

1. When InSocketManager in the Dispatcher pipeline receives a request from the IN, it calls the DAT_ConnectionMonitor module to schedule a timer for the connection.

   **Note:** You can change the timeout value. See "DAT_ConnectionMonitor" in BRM Configuring Pipeline Rating and Discounting for more information.

2. If the request is a DWR message, InSocketManager maps the request to an EDR container and flags the EDR for a DWA message to be returned to the IN.

3. The IN Output pipeline converts the EDR to a DWA message and sends it to the IN.

   **Important:** If AAA Gateway Manager does not send a DWA message within a specified time period, you must ensure that the client handles the failure by closing the connection and failing over to a secondary AAA Gateway Manager.

4. DAT_ConnectionMonitor resets the timer for the connection.

5. If no AAA request or DWR message is received from the IN during the idle timeout period for a connection, DAT_ConnectionMonitor sends a dummy EDR for a DWR message to the IN Output pipeline.

   **Note:** The ISC_PopulateDWRMessage iScript module in the IN Output pipeline generates the DWR message.

6. OutSocketManager sends the DWR message in Diameter protocol format to the IN.

7. If AAA Gateway Manager receives a DWA message from the IN within the idle timeout period, it determines that the connection is working.
If AAA Gateway Manager does not receive a DWA message from the IN, it determines that there is a transport failure and closes the connection to the IN.

**Specifying the Maximum Number of CM Connections for AAA Gateway**

You can determine the maximum number of connections that can be generated by AAA Gateway to the CM by using the following formula:

\[ \text{NumberOfInstancesProcess} + \text{NumberOfInstancesReplay} + \text{IdleConnectionBuffer} \]

where:

- \( \text{NumberOfInstancesProcess} \) is the value of the \( \text{NumberOfInstances} \) registry entry set in the "Processing Pipeline".
  
  See "Configuring Multiple Instances of a Pipeline" in *BRM System Administrator’s Guide* for more information.

- \( \text{NumberOfInstancesReplay} \) is the value of the \( \text{NumberOfInstances} \) registry entry set in the "Replay Pipeline".
  
  See "Configuring Multiple Instances of a Pipeline" in *BRM System Administrator’s Guide* for more information.

- \( \text{IdleConnectionBuffer} \) is the value of the DAT_IdleConnectionBuffer registry entry in the DAT_ConnectionPool module.

To specify the maximum number of connections by AAA Gateway to the CM, edit the \( \text{NumberOfInstances} \) and \( \text{IdleConnectionBuffer} \) entries in your registry file.

**Rebalancing CM Connections**

You can rebalance connections to CMs after one or more CMs is taken offline and brought back up. For example, if you have four CMs and one CM goes offline, the load is automatically balanced among the three remaining CMs. When you bring the fourth CM back on line, the load is not automatically rebalanced. You use the *pin_ctl* utility to rebalance the CMs.

To rebalance CMs:

1. Go to the `BRM_home/bin` directory.
2. Run the *pin_ctl* utility with the `snmpset rebalanceAaaGw` parameter:

   ```
   pin_ctl snmpset rebalanceAaaGw aaa
   ```

   See "*pin_ctl*" in *BRM System Administrator’s Guide* for more information.

To verify that the CM connections were rebalanced, use the SNMP Web interface. See *BRM System Administrator’s Guide* for more information.

Figure 1–2 shows the CM connections:
To troubleshoot AAA Gateway Manager, you configure log files at different points in the pipeline processes. You check them to identify specific problems and fix them. You can configure four kinds of log files to identify problems in different areas of AAA Gateway Manager:

- Using the Process Log
- Using the Pipeline Log
- Using the Stream Log
- Using the EDR Log

**Note:** If AAA Gateway Manager can’t write to log files because of a system error, it logs the message to the standard error, `cerr`. The messages then appear on the console window. You can redirect standard error to a file and read the messages in the file to find the system error.

### Enabling OMF Instrumentation Data Collection in AAA Gateway Manager

You can use the Operations Management Framework (OMF) to get instrumentation data for AAA Gateway Manager. See "About Operations Management Framework" in BRM System Administrator’s Guide for more information. You can enable or disable instrumentation data collection in each AAA Gateway Manager registry file. See "Enabling SNMP Instrumentation Data Collection" in BRM System Administrator’s Guide for more information.

### Getting Diagnostic Data from AAA Gateway Manager

Use the Diagnostic Data Handler to collect analysis data during a crash or critical error, or by performing a snapshot of the system resources. You can enable or disable the Diagnostic Data Handler in each AAA Gateway Manager registry file. See "Using the Diagnostic Data Handler to Get OMF Diagnostic Data" in BRM System Administrator’s Guide for more information.
Using the Process Log

This file contains information about the status of all the AAA Gateway processes. You can set the following log levels depending on the information you want to capture; for example, all process information or only serious errors:

- Normal, which is the default
- Major
- Minor
- Warning
- Critical

You can also specify whether to indicate the specific process for which the logged information applies by setting the **ShowOriginator** entry to True.

---

**Important:** Configure this log before all the pipelines in the registry file.

---

You can also use a Web browser to view the status of the Processing pipeline. See "Viewing the Status of the Processing Pipelines Through the Web" for more information.

**Sample Registry Entry**

```plaintext
ProcessLog
{
    ModuleName = LOG
    Module
    {
        ITO
        {
            FilePath = ./log
            FileName = processSIMPLE
            FileSuffix = .log
            ProcessName = ifw
            MessageGroup = IFW
            ShowOriginator = True
            LogLevel = Critical
        }
        Buffer
        {
            Size = 1000
        }
    }
}
```

Using the Pipeline Log

The pipeline log is a buffer to log specific information about a pipeline. Each pipeline has its own log file with information about that pipeline.

**Sample Registry Entry**

```plaintext
PipelineLog
{
    ModuleName = LOG
```
Module
{
    ITO
    {
        FilePath = ./samples/simple/data/log/pipeline
        FileName = testDispatcherPipeline.log
        ProcessName = ifw
        MessageGroup = IFW
        ShowOriginator = True
        SuppressErrors
            {
                ERR_SPEC_VERSION_INVALID
                ERR_RELEASE_VERSION_INVALID
            }
    }
    Buffer
    {
        Size = 1000
    }
}

Using the Stream Log

The stream log contains detailed messages for one output stream; one stream log file is generated per input stream. The log file contains all error messages for the stream and event; for example, zone data not found.

---

**Important**: The number of stream log files grows indefinitely, so you should delete them periodically to save disk space.

---

Sample Registry Entry

OutputLog
{
    ModuleName = LOG
    Module
    {
        ITO
        {
            FilePath = /ifw/log/stream
            FilePrefix = stream_
            FileSuffix = .log
            LogLevel = normal
            SuppressErrors
                {
                    ERR_SPEC_VERSION_INVALID
                    ERR_RELEASE_VERSION_INVALID
                }
        }
    }
}

Using the EDR Log

You can collect information about the EDRs by configuring the EDRTrace entry in the registry file to write the contents of EDRs at various points in the pipeline to a file. You
can read the EDR dump to verify if an EDR is populated with the output of the appropriate pipeline and if the EDR output is correct.

For example, if the input grammar cannot translate the input data because the data is not mapped correctly in the grammar description, InSocketManager discards the input data and does not generate an EDR. You can configure EDRTrace to dump the EDR after InSocketManager to check if an EDR is generated. The EDR dump has information about the particular fields that are problematic.

Or you can configure EDRTrace to dump the EDR after the FCT_Opcode module. When an EDR is sent to the CM for processing and an error occurs in any component in the BRM server, the error is propagated in an error buffer to the EDR with all the debugging information. You can check the EDR dump to see if the EDR contains an output block or an error buffer block; if it contains an error buffer block, find out what the error is.

See "Troubleshooting Pipeline Modules" in BRM System Administrator’s Guide for more information on configuring EDRTrace to write contents of EDRs to a file.

Viewing the Status of the Processing Pipelines Through the Web

You can also monitor AAA pipeline performance. See "Monitoring and Maintaining Your BRM System" in BRM System Administrator’s Guide for more information.

The information about queues that are shared by multiple Processing pipelines are recorded in the process log. You can access the process log by using your Web browser to monitor the status of the Processing pipelines.

See "Using the Process Log" for more information about the process log.

1. To enable monitoring of the AAA Gateway processes through a Web browser, add the following entry to the appropriate *.reg file:

   ```
   Web
   {
   WebServerPort = port_number
   FileServerAddress = file_server_path
   }
   ```

2. Save and close the file.

3. To view the status of processes, point your browser to the file server.

   **Tip:** Set the process loop timeout (ProcessLoopTimeout) in the *.reg file to 10 to log the status of the queues every 10 seconds.

Using the AAA Pinlog

The AAA Gateway includes the AAA.pinlog file to record messages about flists sent to or received from the CM. The file is located in the /log subdirectory of the AAA Gateway installation directory.

You can set the logging level. See "Setting the Reporting Level for Logging Messages" in BRM System Administrator’s Guide for more information.

About Measuring the Latencies of AAA Requests to Monitor Performance

You can measure the latencies for processing AAA requests in the Diameter protocol and monitor the performance of AAA Gateway Manager by enabling the monitoring
feature. See "Enabling AAA Gateway Manager to Measure Latencies of AAA Requests" for more information.

---

**Note:** Measuring the latency of the AAA requests affects the performance of AAA Gateway Manager, so disable monitoring during normal processing when you are not troubleshooting AAA Gateway Manager. By default, this feature is disabled.

---

By default, enabling monitoring measures the latencies of the following opcodes, which process authorization, reauthorization, and accounting requests:

- PCM_OP_GSM_AUTHORIZE
- PCM_OP_GSM_REAUTHORIZE
- PCM_OP_GSM_STOP_ACCOUNTING
- PCM_OP_TCF_AAA_AUTHORIZE
- PCM_OP_TCF_AAA_REAUTHORIZE
- PCM_OP_TCF_AAA_STOP_ACCOUNTING

You can add opcodes to or delete opcodes from the default list. See "Specifying the AAA Requests to Monitor" for more information.

By default, the latencies are recorded in a latency report file in the `Pipeline_home/formatDesc/Formats/Diameter/log/dump` directory. For information about the report file name and the format of its contents, see "Format of the AAA Requests Latency Report File" for more information.

By default, AAA Gateway Manager logs 1000 records per file and 10 records per write operation. You can configure the number of records per file, how many records to write each time a file is opened, and the location of the report file. See "Configuring the Latency Report File Content" for more information.

---

**Enabling AAA Gateway Manager to Measure Latencies of AAA Requests**

To monitor performance, enable AAA Gateway Manager to record the time taken to process each request:

1. Stop AAA Gateway Manager.
2. Open the `Diameter_v01_OutGrammar.dsc` file in the `Pipeline_home/formatDesc/Formats/Diameter` directory.
3. Set the `enableMonitoringSoln` parameter to `true`:
   ```
   enableMonitoringSoln = true
   ```
4. Save and close the file.
5. Restart AAA Gateway Manager.

---

**Specifying the AAA Requests to Monitor**

To monitor a different set of AAA requests from the default set:

1. Stop AAA Gateway Manager.
2. Open the `Diameter_v01_OutGrammar.dsc` file in the `Pipeline_home/formatDesc/Formats/Diameter` directory.
3. In the `monitorSolution` function block, edit the switch-case statements:
   a. Comment out statements corresponding to the opcodes that you don’t want to monitor.
   b. Add statements for additional opcodes that you want to monitor.

4. Save and close the file.

5. Restart AAA Gateway Manager.

**Format of the AAA Requests Latency Report File**

The latency report for the request is recorded in a file in the `Pipeline_home/formatDesc/Formats/Diameter/logdump` directory. The file name has the following format:

```
opcode_load.event.opc_file_timestamp.opc_file_serial_number
```

where:

- `opcode_load.event` describes the event as opcode processing.
- `opc_file_timestamp` is the time, in the format `YYYYMMDDhhmmss`, when the AAA Gateway Manager started.
- `opc_file_serial_number` is the count of the latency report file.

---

**Note:** When a file reaches the maximum number of records specified in the `recordsPerFile` variable in the `Diameter_v01_OutGrammar.dsc` file, a new file is created.

---

For example:

```
opcode_load.event.20070326130730.61
```

Each event record has six fields in the following format:

```
Event: OP-CODE opcode_name phone_number opcode_thread_id latency start_time_offset_in_ms
```

where:

- `Event: OP-CODE` specifies that the current event is an opcode process.
- `opcode_name` is the name of the opcode called to process a request.
- `phone_number` and the `opcode_thread_id` contain dummy values.
- `latency` is the time in milliseconds taken to process the request.
- `start_time_offset_in_ms` is relative to `opc_file_timestamp` in the latency report file name and is used to reconstruct the exact time that AAA Gateway Manager received a request.

For example:

```
Event: OP-CODE PCM_OP_GSM_AUTHORIZE 123456 1 273 1522790
```

**Configuring the Latency Report File Content**

To configure the latency report file content:

1. Stop AAA Gateway Manager.
2. Open the Diameter_v01_OutGrammar.dsc file in the Pipeline_home/formatDesc/Formats/Diameter directory.

3. Change the default value of any of the following variables:
   - `recordsPerFile` variable to specify the number of records you want in each latency report file.
   - `recordsPerWrite` variable to specify the number of records to write in a file each time it is opened.
   - `eventLogDir` variable to specify a different directory for the latency report file.

4. Save and close the file.

5. Restart AAA Gateway Manager.
AAA Gateway Manager Protocol Support

This document describes the network message protocols supported by Oracle Communications Billing and Revenue Management (BRM) AAA Gateway Manager. For background information, see the following topics:

- Using BRM AAA Gateway Manager
- AAA EDR Container Description

About AAA Gateway Manager Protocol Support

To convert messages from prepaid networks into a format that the pipeline-based AAA Gateway Manager and BRM understand, the Gateway’s InSocketManager passes the messages to a parser. The parser uses protocol-specific input grammar and mapping files to convert the messages into AAA EDR containers, the data structure understood by pipeline function modules. The FCT_Opcode module then converts the data in the containers into an input flist and passes it to the appropriate opcode. The opcode uses the data to perform its normal BRM operations. After the opcode generates an output flist, the FCT_Opcode module converts the flist into an AAA EDR container. The Gateway’s OutSocketManager then passes the container to a parser, which uses protocol-specific output grammar and mapping files to convert the container into an appropriately formatted network response.

Figure 2–1 illustrates the AAA Gateway Manager protocol conversion process. Blue files are used to convert incoming messages to AAA EDR containers. Yellow files are used to convert AAA EDR containers to outgoing messages.
AAA Gateway Manager supports messages based on the following protocols:

- **Diameter protocol**
  
  To enable AAA Gateway Manager to support this protocol, you must install Diameter Charging Manager (see "Installing Diameter Charging Manager"). See "Diameter Protocol Support Files" for information about the grammar and mapping files for this protocol.

- **Tag/LengthOfValue/Value (TLV) format used by the HP OpenCall Message Based Interface (MBI) protocol**
  
  To enable AAA Gateway Manager to support this protocol, you must install HP OpenCall Manager (see "Installing HP OpenCall Manager"). See "MBI Protocol Support Files" for information about the grammar and mapping files for this protocol.

- **MBI CDR protocol**
  
  To enable AAA Gateway Manager to process MBI CDRs, you must install HP OpenCall Manager (see "Installing HP OpenCall Manager"). See "MBI CDR Protocol Support Files" for information about the grammar and mapping files for this protocol.

**Diameter Protocol Support Files**

To process Diameter-based messages, AAA Gateway Manager uses these files:

- Diameter Input Grammar
- Diameter Input Mapping
- Diameter Stream Format Description
Diameter Input Grammar

The Diameter input grammar file (Pipeline_home/formatDesc/Formats/Diameter/Diameter_v01_InGrammar.dsc) contains instructions for generating AAA EDR containers and filling them with data from incoming Diameter-based messages. At startup, AAA Gateway Manager uses this grammar to create a parser that implements the instructions.

By default, the Diameter input grammar supports these opcodes:

- PCM_OP_TCF_AAA_AUTHENTICATE
- PCM_OP_TCF_AAA_AUTHORIZE
- PCM_OP_TCF_AAA_UPDATE_ACCOUNTING
- PCM_OP_TCF_AAA_REALLOCATE
- PCM_OP_TCF_AAA_UPDATE_AND_REALLOCATE
- PCM_OP_TCF_AAA_CANCEL_REALLOCATE
- PCM_OP_TCF_AAA_STOP_REALLOCATE

Diameter Field Blocks

Diameter messages consist of a message header followed by a list of attribute/value pairs (AVPs). The AVPs can be listed in any order. Each header contains a command code to identify the message type. Table 2–1 shows the diameter messages.

Table 2–1  Diameter Messages

<table>
<thead>
<tr>
<th>Diameter Message Type</th>
<th>Command Name</th>
<th>Command Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorize Authenticate</td>
<td>AA-Request (AAR)</td>
<td>265</td>
</tr>
<tr>
<td></td>
<td>AA-Answer (AAA)</td>
<td></td>
</tr>
<tr>
<td>Accounting</td>
<td>Accounting-Request (ACR)</td>
<td>271</td>
</tr>
<tr>
<td></td>
<td>Accounting-Answer (ACA)</td>
<td></td>
</tr>
<tr>
<td>Capabilities Exchange</td>
<td>Capabilities-Exchange-Request (CER)</td>
<td>257</td>
</tr>
<tr>
<td></td>
<td>Capabilities-Exchange-Answer (CEA)</td>
<td></td>
</tr>
<tr>
<td>Credit Control</td>
<td>Credit-Control-Request (CCR)</td>
<td>272</td>
</tr>
<tr>
<td></td>
<td>Credit-Control-Answer (CCA)</td>
<td></td>
</tr>
<tr>
<td>Disconnecting Peer Connections</td>
<td>Disconnect-Peer-Request (DPR)</td>
<td>282</td>
</tr>
<tr>
<td></td>
<td>Disconnect-Peer-Answer (DPA)</td>
<td></td>
</tr>
<tr>
<td>Session Termination</td>
<td>Session-Termination-Request (STR)</td>
<td>275</td>
</tr>
<tr>
<td></td>
<td>Session-Termination-Answer (STA)</td>
<td></td>
</tr>
<tr>
<td>Transport Failure Detection</td>
<td>Device-Watchdog-Request (DWR)</td>
<td>282</td>
</tr>
<tr>
<td></td>
<td>Device-Watchdog-Answer (DWA)</td>
<td></td>
</tr>
</tbody>
</table>
In the Diameter input grammar, the Diameter header is defined as a fixed block, and each field that corresponds to an AVP is defined as a separate block. Each AVP block contains a field descriptor in one of the following formats (the field descriptor is defined in the "Diameter Stream Format Description" file):

- **DiameterType1**(AVP_Code, Flags, Vendor_ID)
- **DiameterType2**(AVP_Code, Flags, Vendor_ID, Size)
- **DiameterType3**(AVP_Code, Flags, Vendor_ID, Zone)

where:
- **Type1** is data type **String** or **Grouped**.
- **Type2** is data type **Integer** or **Decimal**.
- **Type3** is data type **Time**.
- **AVP_Code** is a value defined by the Internet Assigned Numbers Authority (IANA) that uniquely identifies the message type.
  - See "Supported Diameter Messages" for more information.
- **Flags** is one or more of the flags in Table 2–2.

### Table 2–2  Flags

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Indicates that the AVP must be supported. This flag is presumed to apply to all AVPs in incoming messages.</td>
</tr>
<tr>
<td>P</td>
<td>Indicates that the AVP must be encrypted for end-to-end security</td>
</tr>
<tr>
<td>V</td>
<td>Indicates that the AVP header contains the optional Vendor-ID field</td>
</tr>
</tbody>
</table>

- **Vendor_ID** is a value used with the **AVP_Code** to create a vendor-specific Diameter AVP. For all incoming messages, **Vendor_ID** is presumed to be 0.
- **Size** is 32 or 64.
- **Zone** is **local** or **utc**.

### Error Handling

The grammar calls the **edrAddError** function to report errors, such as the failure to create a data block.

These errors are recorded in the AAA Gateway Manager stream log (*Pipeline_*

*home/log/stream/InOutputlog_timestamp_transactionID.log*). The block in which the error occurred is specified in the log.

### Diameter Input Mapping

To transfer data from fields in incoming Diameter-based messages to the appropriate fields in AAA EDR containers, the parser follows the mapping in the Diameter input mapping file (*Pipeline_*

*home/formatDesc/Formats/Diameter/Diameter_v01_InMap.dsc*).
Some AVPs are mapped to different fields in different types of Diameter request messages. For such AVPs, the input mapping file includes multiple subblocks. For example, the USER_NAME AVP block has four request-specific subblocks - AUTHORIZE_AUTHENTICATE, SESSION_TERMINATE, CREDIT_CONTROL, and ACCOUNTING:

USER_NAME
{
AUTHORIZE_AUTHENTICATE
{
    USER_NAME -> DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.DIA_DETAIL.AUTHORIZE_AUTHENTICATE.USER-NAME;
}
SESSION_TERMINATE
{
    USER_NAME -> DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.DIA_DETAIL.SESSION-TERMINATE.USER-NAME;
}
CREDIT_CONTROL
{
    USER_NAME -> DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.DIA_DETAIL.CREDIT-CONTROL.USER-NAME;
}
ACCOUNTING
{
    USER_NAME -> DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.DIA_DETAIL.DIAMETER_AC.USER-NAME;
}
}

When processing a Diameter-based message, the input grammar uses the AVP subblock that corresponds to the request type identified in a message’s header. See "Supported Diameter Messages" for more information about Diameter request messages.

Diameter Stream Format Description

The Diameter stream format description file (Pipeline_home/formatDesc/Formats/Diameter/Diameter_v01.dsc) contains descriptions of the fields in Diameter-based messages supported by AAA Gateway Manager. For example, the following entry specifies that the data type of the HOST_IP_ADDRESS field is DiameterString, its AVP code is 257, it must be supported per the mandatory flag M, and its vendor ID is 0:

HOST_IP_ADDRESS   DiameterString(257, "M", 0);

See "Diameter Field Blocks" for more information about field descriptors.

The input parser uses this file to identify the data type of fields in incoming request messages. The output parser uses this file to format answer messages.

To process Diameter-based network messages, AAA Gateway Manager supports several Diameter data types. See "Diameter Data Types" in BRM Configuring Pipeline Rating and Discounting for more information.
**Diameter Output Grammar**

The Diameter output grammar file (Pipeline_home/formatDesc/Formats/Diameter/Diameter_v01_OutGrammar.dsc) contains instructions for generating outgoing Diameter-based messages and filling them with data from AAA EDR containers. At startup, AAA Gateway Manager uses this grammar to create a parser that implements the instructions.

The output grammar file contains a block for each AAA function and blocks for DWR and DWA messages; only the function blocks associated with the request identified in the message header are incorporated into the parser. Each function block contains the following blocks:

- **Mandatory fields** - The edrOutputMapToBuffer function transfers the data in mandatory fields to a buffer. When BRM finishes processing a request, the outputWrite() function transfers the data in all such function-block buffers into an outgoing Diameter-based answer message.

- **Optional fields** - Before data in optional fields is transferred to a buffer, the edrInternalState function verifies that the fields are included in the output flists of the opcodes that implemented the request or in the protocol-specific data in the ASSOCIATED_PROTOCOL_INFO block.

**Error Handling**

The logPipeline function writes all errors to the pipeline log.

**Diameter Output Mapping**

To transfer data from fields in AAA EDR containers to the appropriate fields in outgoing Diameter-based messages, the parser follows the mapping in the Diameter output mapping file (Pipeline_home/formatDesc/Formats/Diameter/Diameter_v01_OutMap.dsc).

**Supported Diameter Messages**

By default, AAA Gateway Manager supports these Diameter NASREQ application messages:

- **Device Watchdog messages**
  - **Device Watchdog Request** - Request to check the status of the connection between the IN and AAA Gateway Manager when there is no AAA message exchange.
  - **Device Watchdog Answer** - Response for the DWR message from AAA Gateway Manager to the IN and from the IN to AAA Gateway Manager.

- **Authentication and authorization**
  - **AA-Request** - Request for authentication and authorization. Mapped to the input flist of PCM_OP_TCF_AAA_AUTHORIZE. If authentication succeeds, AAA Gateway Server sends an AA-Answer message with the Result-Code AVP set to DIAMETER_SUCCESS to the Diameter server. If authentication fails, the Result-Code AVP is set to the appropriate error code.
  - **AA-Answer** - Answer to request for authentication and authorization.

- **Credit control**
  - **CC-Request** - Request for credit control operation. There are four types of CC-Request messages. The type is specified in the message’s CC-Request-Type
AVPs. They are mapped to the input list of the BRM opcodes of Table 2–3.

### Table 2–3  BRM Opcodes

<table>
<thead>
<tr>
<th>Message Type</th>
<th>Description</th>
<th>Mapped To</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITIAL_REQUEST</td>
<td>Used to start a credit control session. Contains credit control information</td>
<td>PCM_OP_TCF_AAA_AUTHORIZE</td>
</tr>
<tr>
<td></td>
<td>relevant to the initiation of the session.</td>
<td></td>
</tr>
<tr>
<td>UPDATE_REQUEST</td>
<td>Contains credit control information for an existing credit control session.</td>
<td>PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE</td>
</tr>
<tr>
<td></td>
<td>Should be sent every time a credit control reauthorization is needed when an</td>
<td></td>
</tr>
<tr>
<td></td>
<td>allocated resource or validity time expires</td>
<td></td>
</tr>
<tr>
<td>TERMINATION_REQUEST</td>
<td>Used to end a credit control session. Contains credit control information</td>
<td>When accounting is disabled in the Diameter</td>
</tr>
<tr>
<td></td>
<td>relevant to the terminated session.</td>
<td>registry file (diameter_charge.reg), this</td>
</tr>
<tr>
<td></td>
<td></td>
<td>message type is mapped to PCM_OP_TCF_AAA_STOP_</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACCOUNTING.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When accounting is enabled in the Diameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>registry file, this message type is not</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mapped to any opcode. Instead, the opcode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>call is skipped, and a dummy CC-Answer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>message is generated.</td>
</tr>
<tr>
<td></td>
<td>Note: By default, accounting is disabled. If the Diameter credit control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>application uses accounting commands, you must enable accounting by adding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the ifw.isAccountingEnabled entry to the Diameter registry file and setting</td>
<td></td>
</tr>
<tr>
<td></td>
<td>it to 1. See &quot;Configuring AAA Gateway Manager&quot; for more information.</td>
<td></td>
</tr>
<tr>
<td>EVENT_REQUEST</td>
<td>Used when there is no need to maintain any credit control session state in</td>
<td>PCM_OP_TCF_AAA_STOP_ACCOUNTING</td>
</tr>
<tr>
<td></td>
<td>the server.</td>
<td>The following event requests are supported:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>direct debit, query balance, and service price</td>
</tr>
<tr>
<td></td>
<td></td>
<td>enquiry.</td>
</tr>
</tbody>
</table>

### Table 2–4  Supported AVPs for CCR Initial Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session-Id</td>
<td>M</td>
<td>AUTHORIZATION_ID</td>
<td>String</td>
<td>Single</td>
<td>Mapped 1</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>ORIGIN_NETWORK</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Destination-Realm</td>
<td>M</td>
<td>DESTINATION_NETWORK</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Auth-Application-Id</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed 2</td>
</tr>
<tr>
<td>Service-Context-Id</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>CC-Request-Type</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Request-Number</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Destination-Host</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>User-Name</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>CC-Sub-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
</tbody>
</table>
Table 2–5 lists the supported AVPs for CC-Request (CCR) Update messages:

Table 2–5  Supported AVPs for CCR Update Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/ Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acct-Multi-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Event-Timestamp</td>
<td>O</td>
<td>START_T</td>
<td>Time</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Subscription-Id</td>
<td>M</td>
<td>IMSI/MSISDN and MSID</td>
<td>Grouped</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Service-Identifier</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Termination-Cause</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped 3</td>
</tr>
<tr>
<td>Requested-Service-Unit</td>
<td>O</td>
<td>If Requested-Service-Unit is present, one of the following fields will be populated, depending on the input: QUANTITY, REQ_BYTES_UPLINK, OR REQ_BYTES_DOWNLINK</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Requested-Action</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Used-Service-Unit</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Multiple-Services-Indicator</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Multiple-Services-Credit-Control</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Service-Parameter-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>CC-Correlation-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>User-Equipment-Info</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Proxy-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Route-Record</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>AVP</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
</tbody>
</table>

1 Mapped - The field is used by AAA Gateway Manager.
2 Parsed - The field is parsed but not used by AAA Gateway Manager.
3 Not mapped - The field is not used by AAA Gateway Manager.
Table 2–6 lists the supported AVPs for CC-Request (CCR) Final messages:

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC-Request-Number</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Destination-Host</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>User-Name</td>
<td>O</td>
<td>MSID</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Sub-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Acct-Multi-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Event-Timestamp</td>
<td>O</td>
<td>END_T</td>
<td>Time</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Subscription-Id</td>
<td>M</td>
<td>IMSI/MSISDN and MSID</td>
<td>Grouped</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Service-Identifier</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Termination-Cause</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped 3</td>
</tr>
<tr>
<td>Requested-Service-Unit</td>
<td>O</td>
<td>If Requested-Service-Unit is present, one of the following fields will be populated, depending on the input: QUANTITY, REQ_BYTES_UPLINK, OR REQ_BYTES_DOWNLINK</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Requested-Action</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Used-Service-Unit</td>
<td>O</td>
<td>QUANTITY/USED_BYTES_UPLINK/USED_BYTES_DOWNLINK</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Multiple-Services-Indicator</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Multiple-Services-Credit-Control</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Service-Parameter-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>CC-Correlation-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>User-Equipment-Info</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Proxy-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Route-Record</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>AVP</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
</tbody>
</table>

1 Mapped - The field is used by AAA Gateway Manager.
2 Parsed - The field is parsed but not used by AAA Gateway Manager.
3 Not mapped - The field is not used by AAA Gateway Manager.
**Table 2–6 Supported AVPs for CCR Final Messages**

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/ Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session-Id</td>
<td>M</td>
<td>AUTHORIZATION_ID</td>
<td>String</td>
<td>Single</td>
<td>Mapped ¹</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>ORIGIN_NETWORK</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Destination-Realm</td>
<td>M</td>
<td>DESTINATION_NETWORK</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Auth-Application-Id</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed ²</td>
</tr>
<tr>
<td>Service-Context-Id</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>CC-Request-Type</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Request-Number</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Destination-Host</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>User-Name</td>
<td>O</td>
<td>MSID</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Sub-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Acct-Multi-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Event-Timestamp</td>
<td>O</td>
<td>END_T</td>
<td>Time</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Subscription-Id</td>
<td>M</td>
<td>IMSI/MSISDN and MSID</td>
<td>Grouped</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Service-Identifier</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Termination-Cause</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Requested-Service-Unit</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped ³</td>
</tr>
<tr>
<td>Requested-Action</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Used-Service-Unit</td>
<td>O</td>
<td>QUANTITY/USED_BYTES_UPLINK/USED_BYTES_DOWNLINK</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Multiple-Services-Indicator</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Multiple-Services-Credit-Control</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Service-Parameter-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>CC-Correlation-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>User-Equipment-Info</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Proxy-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Route-Record</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>AVP</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
</tbody>
</table>

1 Mapped - The field is used by AAA Gateway Manager.
2 Parsed - The field is parsed but not used by AAA Gateway Manager.
3 Not mapped - The field is not used by AAA Gateway Manager.

- **CC-Answer** - Answer to request for credit control operation.

Table 2–7 lists the supported AVPs for CC-Answer (CCA) Initial messages:
Table 2–7  Supported AVPs for CCA Initial Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/ Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session-Id</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped 1</td>
</tr>
<tr>
<td>Result-Code</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-Host</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Auth-Application-Id</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped 2</td>
</tr>
<tr>
<td>CC-Request-Type</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Request-Number</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>User-Name</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Session-Failover</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>CC-Sub-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Acct-Multi-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Event-Timestamp</td>
<td>O</td>
<td>NA</td>
<td>Time</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Granted-Service-Unit</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Mapped</td>
</tr>
<tr>
<td>Multiple-Services-Credit-Control</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Cost-Information</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Final-Unit-Indication</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Check-Balance-Result</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Credit-Control-Failure-Handling</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Direct-Debiting-Failure-Handling</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Validity-Time</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Host</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Host-Usage</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Max-Cache-Time</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Proxy-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Route-Record</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Failed-AVP</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>AVP</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

1  Mapped - The field is used by AAA Gateway Manager.
2  Not mapped - The field is not used by AAA Gateway Manager.

Table 2–8 lists the supported AVPs for CC-Answer (CCA) Update messages.
### Table 2–8  Supported AVPs for CCA Update Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/ Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session-Id</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Not mapped 1</td>
</tr>
<tr>
<td>Result-Code</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped 2</td>
</tr>
<tr>
<td>Origin-Host</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Auth-Application-Id</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>CC-Request-Type</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Request-Number</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>User-Name</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Session-Failover</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>CC-Sub-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Acct-Multi-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Event-Timestamp</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Granted-Service-Unit</td>
<td>O</td>
<td>BALANCES</td>
<td>NA</td>
<td>NA</td>
<td>Mapped</td>
</tr>
<tr>
<td>Multiple-Services-Credit-Control</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Cost-Information</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Final-Unit-Indication</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Check-Balance-Result</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Credit-Control-Failure-Handling</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Direct-Debting-Failure-Handling</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Validity-Time</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Host</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Host-Usage</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Max-Cache-Time</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Proxy-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Route-Record</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Failed-AVP</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
</tbody>
</table>

1 Not mapped - The field is not used by AAA Gateway Manager.
2 Mapped - The field is used by AAA Gateway Manager.

Table 2–9 lists the supported AVPs for CC-Answer (CCA) Final messages:
Table 2–9  Supported AVPs for CCA Final Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session-Id</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Not mapped 1</td>
</tr>
<tr>
<td>Result-Code</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped 2</td>
</tr>
<tr>
<td>Origin-Host</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Auth-Application-Id</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>CC-Request-Type</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Request-Number</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>User-Name</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Session-Failover</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>CC-Sub-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Acct-Multi-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Event-Timestamp</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Granted-Service-Unit</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>Mapped</td>
</tr>
<tr>
<td>Multiple-Services-Credit-Control</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Cost-Information</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Final-Unit-Indication</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Check-Balance-Result</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Credit-Control-Failure-Handling</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Direct-Debiting-Failure-Handling</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Validity-Time</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Host</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Host-Usage</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Max-Cache-Time</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Proxy-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Route-Record</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Failed-AVP</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>AVP</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

1  Not mapped - The field is not used by AAA Gateway Manager.
2  Mapped - The field is used by AAA Gateway Manager.

- **Accounting**
  - **AC-Request** - Request for accounting. There are four types of AC-Request messages. The type is specified in the message’s Accounting-Record-Type
AVP. They are mapped to the input flist of the BRM opcodes shown in Table 2–10.

Table 2–10 AC Request BRM Opcodes

<table>
<thead>
<tr>
<th>Message Type</th>
<th>Description</th>
<th>Mapped To</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>Used to start an active session. Includes session start timestamp.</td>
<td>PCM_OP_TCF_AAA_START_ACCOUNTING</td>
</tr>
<tr>
<td>INTERIM</td>
<td>Used to provide usage information about an ongoing active session.</td>
<td>PCM_OP_TCF_AAA_UPDATE_ACCOUNTING</td>
</tr>
<tr>
<td>STOP</td>
<td>Used to terminate an active session.</td>
<td>PCM_OP_TCF_AAA_STOP_ACCOUNTING</td>
</tr>
<tr>
<td>EVENT</td>
<td>Used to record a one-time event (event’s start and end times are the same).</td>
<td>PCM_OP_TCF_AAA_STOP_ACCOUNTING</td>
</tr>
</tbody>
</table>

- AC-Answer - Answer to request for accounting.
- Session termination
  - ST-Request - Request to terminate session. Mapped to the input flist of PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION.
  - ST-Answer - Answer to request to terminate session.
- AAA Transport and Capabilities Exchange

When two Diameter peers establish a connection, they must first exchange Capabilities Exchange messages: Capabilities-Exchange-Request (CER) and Capabilities-Exchange-Answer (CEA). Diameter peers exchange CER and CEA messages to identify the client and its capabilities, such as the client’s IP address, supported Diameter version, supported Diameter applications (authentication, authorization, accounting) and security mechanisms.

When a Diameter client establishes a connection with the AAA Gateway Manager, the InSocketManager input module stores the client information (port number) in the DAT_ConnectionMonitor plug-in and sets the client state to INITIAL. The client must send a CER message when it’s in the INITIAL state to the AAA Gateway Manager within the time interval that you configure. See "Specifying the Time Interval for Closing the Client Connection" for more information.

The ISC_Update_ConnectionMonitor iScript updates the client state from INITIAL to OK if a CER message is received within the time interval. Otherwise, DAT_ConnectionMonitor closes the connection.

The Diameter client should also include all of its supported applications in the CER message. ISC_Update_ConnectionMonitor checks the Auth-Application-Id and Acct-Application-Id fields in the CER message to identify the services needed by the client. If AAA Gateway Manager does not have any applications in common, it returns a CEA message with the Result-Code AVP (attribute-value-pair) set to DIAMETER_NO_COMMON_APPLICATION and closes the connection.

AAA Gateway Manager supports the following values for Auth-Application-Id and Acct-Application-Id:
- Diameter Common Messages 0
- Diameter Base Accounting 3
- Diameter CCA Message 4

Table 2–11 lists the supported AVPs for CER messages.
Table 2–11  Supported AVPs for CER Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/ Mandatory (RFC)</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin-Host</td>
<td>M</td>
<td>String</td>
<td>Single</td>
<td>Mapped 1</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Host-IP-Address</td>
<td>M</td>
<td>String</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Vendor-Id</td>
<td>M</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Product-Name</td>
<td>M</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Supported-Vendor-Id</td>
<td>O</td>
<td>Integer</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Auth-Application-Id</td>
<td>O</td>
<td>Integer</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Inband-Security-Id</td>
<td>O</td>
<td>Integer</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Acct-Application-Id</td>
<td>O</td>
<td>Integer</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Vendor-Specific-Application-Id</td>
<td>O</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Firmware-Revision</td>
<td>O</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>AVP</td>
<td>O</td>
<td>NA</td>
<td>Multiple</td>
<td>NA</td>
</tr>
</tbody>
</table>

1  Mapped - The field is used by AAA Gateway Manager.

Table 2–12 lists the supported AVPs for CEA messages.

Table 2–12  Supported AVPs for CEA Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/ Mandatory (RFC)</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result-Code</td>
<td>M</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped 1</td>
</tr>
<tr>
<td>Origin-Host</td>
<td>M</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Host-IP-Address</td>
<td>M</td>
<td>String</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Vendor-Id</td>
<td>M</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Product-Name</td>
<td>M</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Error-Message</td>
<td>O</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Failed-AVP</td>
<td>O</td>
<td>NA</td>
<td>Multiple</td>
<td>Not mapped 2</td>
</tr>
<tr>
<td>Supported-Vendor-Id</td>
<td>O</td>
<td>Integer</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Auth-Application-Id</td>
<td>O</td>
<td>Integer</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Acct-Application-Id</td>
<td>O</td>
<td>Integer</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Vendor-Specific-Application-Id</td>
<td>O</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Firmware-Revision</td>
<td>O</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>AVP</td>
<td>O</td>
<td>NA</td>
<td>Multiple</td>
<td>NA</td>
</tr>
</tbody>
</table>
Disconnect Peer

A Diameter client exchanges Disconnect-Peer-Request (DPR) and Disconnect-Peer-Answer (DPA) messages to inform its peer of its intent to disconnect. The Diameter client usually initiates DPR messages. When AAA Gateway Manager receives a DPR message from a Diameter client, the ISC_Update_ConnectionMonitor iScript closes the connection.

The AAA Gateway Manager may also send a DPR message to the Diameter client to close the connection when the AAA Gateway server needs to restart or shut down by sending probe to DAT_ConnectionMonitor.

You use the following command to send the probe for AAA Gateway Manager to close client connections:

```
snmpSet <Agent_Host_Name> 1.3.6.1.4.1.3512.1.7.2.1.1.1.34.0 -P<Agent_Port_Number>
```

where:

- `Agent_Host_Name` is the SNMP host name
- `Agent_Port_Number` is the SNMP port number.

For example, the following command sends the probe 1.3.6.1.4.1.3512.1.7.2.1.1.1.34.0 to SNMP host agent1.example.com at SNMP port 23628.

```
snmpSet agent1.example.com 1.3.6.1.4.1.3512.1.7.2.1.1.1.34.0 -P23628
```

SNMP++ Set to agent1.example.com SNMPV1 Retries=1 Timeout=1000ms
SET-community=public GET-community=public
Oid = 1.3.6.1.4.1.3512.1.7.2.1.1.1.1.34.0
Current Value = << WRITE-ONLY PROBE >>
Value Type is Octet String
Please enter new value: DPR
Set Status = Success

**Note:** When AAA Gateway Manager sends DPR message, it does not wait for DPA.

Table 2–13 lists the supported AVPs for DPR messages.

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/Mandatory (RFC)</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin-Host</td>
<td>M</td>
<td>String</td>
<td>Single</td>
<td>Mapped 1</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Disconnect-Cause</td>
<td>M</td>
<td>NA</td>
<td>Single</td>
<td>Parsed 2</td>
</tr>
</tbody>
</table>

1 Mapped - The field is used by AAA Gateway Manager.
2 Parsed - The field is parsed but not used by AAA Gateway Manager.
Table 2–14 lists the supported AVPs for DPA messages.

### Table 2–14 Supported AVPs for DPA Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/Mandatory (RFC)</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result-Code</td>
<td>M</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped ¹</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-Host</td>
<td>M</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Error-Message</td>
<td>O</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Failed-AVP</td>
<td>O</td>
<td>NA</td>
<td>Multiple</td>
<td>Not mapped ²</td>
</tr>
</tbody>
</table>

¹ Mapped - The field is used by AAA Gateway Manager.
² Not mapped - The field is not used by AAA Gateway Manager.

- **Balance Enquiry**

Diameter clients use Balance Enquiry messages to determine the available balance in the end user’s account. For example, Balance Enquiry messages are used to verify that the end user’s available balance would cover the cost of a certain service.

When AAA Gateway Manager receives a Balance Enquiry Credit-Control-Request (CCR) message, it performs the balance check and responds with the Credit-Control-Answer (CCA) message with Check-Balance-Result AVP set to the ENOUGH_CREDIT or NO_CREDIT and the Cost-Information AVP set to the balance.

When a Balance Enquiry message is sent to BRM, the CC-Request-Type value should be set to EVENT_REQUEST and the Requested-Action AVP must be set to CHECK_BALANCE in the CCR message.

Table 2–15 lists the supported AVPs for Balance Enquiry CCR messages:

### Table 2–15 Supported AVPs for Balance Enquiry CCR Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session-Id</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped ¹</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Destination-Realm</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Auth-Application-Id</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed ²</td>
</tr>
<tr>
<td>Service-Context-Id</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>CC-Request-Type</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Request-Number</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Destination-Host</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>User-Name</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Sub-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped ³</td>
</tr>
<tr>
<td>Acct-Multi-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
</tbody>
</table>
### Table 2–16 Supported AVPs for Balance Enquiry CCA Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Event-Timestamp</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Subscription-Id</td>
<td>M</td>
<td>IMSI/MSISDN and MSID</td>
<td>Grouped</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Service-Identifier</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Termination-Cause</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Requested-Service-Unit</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Parsed</td>
</tr>
<tr>
<td>Requested-Action</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Used-Service-Unit</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Parsed</td>
</tr>
<tr>
<td>Multiple-Services-Indicator</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Multiple-Services-Credit-Control</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Service-Parameter-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>CC-Correlation-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>User-Equipment-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>NA</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Proxy-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Route-Record</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>AVP</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
</tbody>
</table>

1 Mapped - The field is used by AAA Gateway Manager.
2 Parsed - The field is parsed but not used by AAA Gateway Manager.
3 Not mapped - The field is not used by AAA Gateway Manager.

Table 2–16 lists the supported AVPs for Balance Enquiry CCA messages:

### Table 2–16 Supported AVPs for Balance Enquiry CCA Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session-Id</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped ¹</td>
</tr>
<tr>
<td>Result-Code</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-Host</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Auth-Application-Id</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Request-Type</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Request-Number</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>User-Name</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Session-Failover</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped ²</td>
</tr>
<tr>
<td>CC-Sub-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>MAPPED</td>
</tr>
<tr>
<td>Acct-Multi-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
</tbody>
</table>

¹ Mapped - The field is used by AAA Gateway Manager.
² Parsed - The field is parsed but not used by AAA Gateway Manager.
Service Price Enquiry

Diameter clients use Service Price Enquiry messages to determine the cost of a certain service. For example, Service Price Enquiry messages are used to obtain the cost of the service before using the service.

When AAA Gateway Manager receives a Service Price Enquiry CCR message, it calculates the cost of the service (does not perform any balance updates or resource reservation) and returns a CCA message with the Cost-Information AVP set to the cost of the service.

When a Service Price Enquiry message is sent to BRM, the CC-Request-Type value should be set to EVENT_REQUEST and the Requested-Action AVP must be set to PRICE_ENQUIRY in the CCR message.

Table 2–17 lists the supported AVPs for Service Enquiry CCR messages.

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Event-Timestamp</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Granted-Service-Unit</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Multiple-Services-Credit-Control</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Cost-Information</td>
<td>O</td>
<td>PIN_FLD_BALANCES</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Final-Unit-Indication</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Check-Balance-Result</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Credit-Control-Failure-Handling</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Direct-Debiting-Failure-Handling</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Validity-Time</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Host</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Host-Usage</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Max-Cache-Time</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Proxy-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Route-Record</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Failed-AVP</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>AVP</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

1 Mapped - The field is used by AAA Gateway Manager.
2 Not mapped - The field is not used by AAA Gateway Manager.

Table 2–16 (Cont.) Supported AVPs for Balance Enquiry CCA Messages
### Table 2–17  Supported AVPs for Service Enquiry CCR Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/ Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session-Id</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped ¹</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Destination-Realm</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Auth-Application-Id</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Service-Context-Id</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Parsed ²</td>
</tr>
<tr>
<td>CC-Request-Type</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Request-Number</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Destination-Host</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>User-Name</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Sub-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped ³</td>
</tr>
<tr>
<td>Acct-Multi-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Event-Timestamp</td>
<td>O</td>
<td>START_T</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Subscription-Id</td>
<td>M</td>
<td>IMSI/MSIS DN</td>
<td>Grouped</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Service-Identifier</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Termination-Cause</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Parsed</td>
</tr>
<tr>
<td>Requested-Service-Unit</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Requested-Action</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Used-Service-Unit</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Multiple-Services-Indicator</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Multiple-Services-Credit-Control</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Service-Parameter-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>CC-Correlation-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>User-Equipment-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>NA</td>
<td>Not mapped</td>
</tr>
</tbody>
</table>

1. Optional
2. Optional/mandatory
3. Optional

---

2-20  BRM AAA Gateway Manager
### Table 2–18 Supported AVPs for Service Enquiry CCA Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/ Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Route-Record</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>AVP</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
</tbody>
</table>

1. Mapped - The field is used by AAA Gateway Manager.
2. Parsed - The field is parsed but not used by AAA Gateway Manager.
3. Not mapped - The field is not used by AAA Gateway Manager.

Table 2–18 lists the supported AVPs for Service Enquiry CCA messages:

### Table 2–18 Supported AVPs for Service Enquiry CCA Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/ Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session-Id</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped 1</td>
</tr>
<tr>
<td>Result-Code</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-Host</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-Realm</td>
<td>M</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Auth-Application-Id</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Request-Type</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Request-Number</td>
<td>M</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>User-Name</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>CC-Session-Failover</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped 2</td>
</tr>
<tr>
<td>CC-Sub-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Acct-Multi-Session-Id</td>
<td>O</td>
<td>NA</td>
<td>String</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Origin-State-Id</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Event-Timestamp</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Granted-Service-Unit</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Multiple-Services-Credit-Control</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Cost-Information</td>
<td>O</td>
<td>BALANCES</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
<tr>
<td>Final-Unit-Indication</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Check-Balance-Result</td>
<td>NA</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Credit-Control-Failure-Handling</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Mapped</td>
</tr>
<tr>
<td>Direct-Debiting-Failure-Handling</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Validity-Time</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Host</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Host-Usage</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Redirect-Max-Cache-Time</td>
<td>O</td>
<td>NA</td>
<td>Integer</td>
<td>Single</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Proxy-Info</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Mapped</td>
</tr>
</tbody>
</table>
Specifying the Time Interval for Closing the Client Connection

Set the DAT_ConnectionMonitor KeepAliveInterval registry entry to a value in milliseconds to specify the time interval to wait for a CER message from the Diameter client. If AAA Gateway Manager does not receive a CER message within this time interval, it closes the client connection.

Note: The KeepAliveInterval value is also used to send Device Watchdog Request (DWR) messages. See the documentation about DAT_ConnectionMonitor.

Diameter Error Messages

There are two types of errors associated with Diameter: protocol errors and application errors. Protocol errors occur at the base protocol level, and application errors generally occur due to a problem with a function specified in a Diameter application. When AAA Gateway Manager receives a request message that causes a protocol error, it returns an answer message with the Result-Code AVP set to the appropriate protocol error value. Similarly, when application errors occur, an answer message is returned with the appropriate error code.

AAA Gateway Manager supports the Diameter error codes shown in Table 2–19.

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/ Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route-Record</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>Failed-AVP</td>
<td>O</td>
<td>NA</td>
<td>Grouped</td>
<td>Multiple</td>
<td>Not mapped</td>
</tr>
<tr>
<td>AVP</td>
<td>O</td>
<td>NA</td>
<td>NA</td>
<td>Multiple</td>
<td>NA</td>
</tr>
</tbody>
</table>

1 Mapped - The field is used by AAA Gateway Manager.
2 Not mapped - The field is not used by AAA Gateway Manager.

Table 2–18 (Cont.) Supported AVPs for Service Enquiry CCA Messages

<table>
<thead>
<tr>
<th>Diameter AVP</th>
<th>Optional/ Mandatory (RFC)</th>
<th>BRM Field</th>
<th>Data Type</th>
<th>Occurrence</th>
<th>AAA Gateway EDR Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diameter Error Code</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAMETER_SUCCESS (2001)</td>
<td>The request was successfully completed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAMETER_COMMAND_UNSUPPORTED (3001)</td>
<td>The request contained a command code that AAA Gateway Manager did not recognize or does not support.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAMETER_TOO_BUSY (3004)</td>
<td>AAA Gateway Manager can’t provide the requested service because the gateway server is busy.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAMETER_AUTHENTICATION_REJECTED (4001)</td>
<td>The authentication process for the end user failed; for example, due to an invalid password.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAMETER_CREDIT_LIMIT_REACHED (4012)</td>
<td>The request for service is denied because the end user’s account has insufficient funds to cover the cost of the service.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAMETER_AUTHORIZATION_REJECTED (5003)</td>
<td>AAA Gateway Manager could not authorize the user for the service requested because the user is not permitted to use the requested service.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAMETER_INVALID_AVP_VALUE (5004)</td>
<td>The request contained an AVP with an invalid value in its data portion. AAA Gateway Manager returns an answer message with the Failed-AVP AVP set to the invalid values.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To process MBI-based messages, AAA Gateway Manager uses these files:

- MBI Input Grammar
- MBI Input Mapping
- MBI Stream Format Description
- MBI Output Grammar
- MBI Output Mapping

### MBI Input Grammar

The MBI input grammar file (`Pipeline_home/formatDesc/Formats/MBI/MBI_v1_InGrammar.dsc`) contains instructions for generating AAA EDR containers and filling them with data from incoming MBI-based messages. At startup, AAA Gateway Manager uses this grammar to create a parser that implements the instructions.

Each incoming MBI-based message begins with a wire header. The first field after the wire header is the MESSAGE_TYPE field. All subsequent fields in the message are mapped to opcode blocks according to the message type specified in this field.

**Note:** Because the subsequent fields can be arranged in any order, each field is defined as a separate block in the grammar.

### Error Handling

The grammar instructs the parser to use the `edrAddError` function to report errors, such as a failure to create a data block.

These errors are recorded in the AAA Gateway Manager stream log (`Pipeline_home/log/stream/InOutputlog_timestamp_transactionID.log`). The block in which the error occurred is specified in the log.

## Table 2–19 (Cont.) Supported Diameter Error Codes

<table>
<thead>
<tr>
<th>Diameter Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIAMETER_MISSING_AVPI (5005)</td>
<td>A request was received that did not contain an AVP required by the Command-Code definition. AAA Gateway Manager returns an answer message with the Failed-AVP AVP set to an example of the missing AVP with the vendor ID, if applicable.</td>
</tr>
<tr>
<td>DIAMETER_NO_COMMON_APPLICATION (5010)</td>
<td>There are no common applications supported between the Diameter peers.</td>
</tr>
<tr>
<td>DIAMETER_UNABLE_TO_COMPLY (5012)</td>
<td>The request is rejected due to an internal error.</td>
</tr>
<tr>
<td>DIAMETER_USER_UNKNOWN (5030)</td>
<td>The specified end user is not found in the AAA Gateway server.</td>
</tr>
<tr>
<td>DIAMETER_RATING_FAILED (5031)</td>
<td>AAA Gateway Manager was unable to rate the service request due to insufficient rating input, incorrect AVP combination, or an AVP value that is not recognized or supported in the rating.</td>
</tr>
</tbody>
</table>
MBI Input Mapping

To transfer data from fields in incoming flist-based messages to the appropriate fields in AAA EDR containers, the parser follows the mapping in the flist input mapping file (Pipeline_home/formatDesc/Formats/MBI/MBI_v1_InMap.dsc).

MBI Stream Format Description

The MBI stream format description file (Pipeline_home/formatDesc/Formats/MBI/MBI_v1.dsc) specifies the data type of the fields in MBI-based messages that are supported by AAA Gateway Manager. For example, this entry specifies that the data type of the CALL_ATMPT_TM field is TlvTimestamp:

```shell
CALL_ATMPT_TM  TlvTimestamp(1029, 2, 2);
```

The input parser uses this file to identify the data type of fields in incoming network messages. The output parser uses this file to format responses.

To process MBI-based network messages, AAA Gateway Manager supports several MBI data types. See "MBI Data Types" in BRM Configuring Pipeline Rating and Discounting for more information.

MBI Output Grammar

The MBI output grammar file (Pipeline_home/formatDesc/Formats/MBI/MBI_v1_OutGrammar.dsc) contains instructions for generating outgoing MBI-based messages and filling them with data from AAA EDR containers. At startup, AAA Gateway Manager uses this grammar to create a parser that implements the instructions.

MBI Output Mapping

To transfer data from fields in AAA EDR containers to the appropriate fields in outgoing MBI-based messages, the parser follows the mapping in the MBI output mapping file (Pipeline_home/formatDesc/Formats/MBI/MBI_v1_OutMap.dsc).

The output grammar file contains a block for each AAA function. Each function block contains the following blocks:

- **Mandatory fields** - The edrOutputMapToBuffer function transfers the data in mandatory fields to a buffer. When BRM finishes processing a request, the outputWrite function transfers the data in all such function-block buffers into an outgoing MBI-based message.

- **Optional fields** - Before data in optional fields is transferred to a buffer, the edrInternalState function verifies that the fields are included in the output flists of the opcodes that implemented the request or in the protocol-specific data in the ASSOCIATED_PROTOCOL_INFO block.

**Error Handling**

The logPipeline function writes all errors to the pipeline log.

MBI Service Types Supported and Their Mapping to BRM Objects

AAA Gateway Manager supports only the MBI service types mapped to BRM service types and events as shown in Table 2–20.
Supported MBI Messages

By default, AAA Gateway supports these MBI-based messages:

- **Authorize_req** - When a customer attempts to initiate a prepaid service, an HP OpenCall Service Execution Processor (SEP) sends this message to AAA Gateway Manager to authorize the service. AAA Gateway Manager calls the PCM_OP_TCF_AAA_AUTHORIZE opcode with the service-specific (GPRS or GSM) information in the PIN_FLD_EXTENDED_INFO field of the input flist.

- **Authorize_conf** - If AAA Gateway Manager verifies that the customer is authorized to use the service and the account balance is sufficient to initiate the service, the gateway sends this message with a token (for example, the time reserved for the service) to the SEP. AAA Gateway Manager calls the PCM_OP_TCF_AAA_AUTHORIZE opcode with the service-specific (GPRS or GSM) information in the PIN_FLD_EXTENDED_INFO field of the input flist for verification.

- **Reauthorize_req** - When a previously authorized customer's token expires or is about to expire, the SEP sends this message to the gateway, which calls the PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode with the service-specific (GPRS or GSM) information in the PIN_FLD_EXTENDED_INFO field of the input flist. The gateway attempts to reserve a new token for the ongoing session. These messages are sent periodically until the customer runs out of credit or ends the session.

- **Reauthorize_conf** - If reauthorization is successful, the gateway sends this message to the SEP.

- **Authorize_reject** - If authorization or reauthorization is unsuccessful, the gateway sends this message to the SEP.

---

**Table 2-20 Supported MBI Service Types**

<table>
<thead>
<tr>
<th>MBI Service Type</th>
<th>BRM Service Type</th>
<th>BRM Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>MO_voice</td>
<td>/service/telco/gsm/telephony</td>
<td>/event/session/telco/gsm</td>
</tr>
<tr>
<td>MT_voice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF_voice_regular</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MO_fax</td>
<td>/service/telco/gsm/fax</td>
<td>/event/session/telco/gsm</td>
</tr>
<tr>
<td>MT_fax</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF_fax</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MO_SMS</td>
<td>/service/telco/gsm/sms</td>
<td>/event/session/telco/gsm</td>
</tr>
<tr>
<td>MT_SMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MO_data</td>
<td>/service/telco/gsm/data</td>
<td>/event/session/telco/gsm</td>
</tr>
<tr>
<td>MT_data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF_data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MO_GPRS</td>
<td>/service/telco/gprs</td>
<td>/event/session/telco/gprs</td>
</tr>
</tbody>
</table>

**Note:** USSD_Call_Back is not fully supported. The Cmd_ind and Cmd_res messages are mapped to EDR container fields. You can customize the grammar files to call custom opcodes that you write to process these messages.
- **End_req** - When a customer hangs up or a session is not completed, the SEP sends this message to the gateway to end and rate the session. AAA Gateway Manager calls the PCM_OP_TCF_AAA_STOP_ACCOUNTING opcode with the service-specific (GPRS or GSM) information in the PIN_FLD_EXTENDED_INFO field of the input flist.

  If the reason for terminating the call is DESTINATION_BUSY, NO_ANSWER, CALLER_ABANDON, or SMS_FAILURE, PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION is called.

- **End_ack** - After BRM rates the session, debits the customer’s balance, and saves the session data, the gateway sends this message with the customer’s remaining account balance to the SEP. AAA Gateway Manager calls the PCM_OP_TCF_AAA_STOP_ACCOUNTING opcode with the service-specific (GPRS or GSM) information in the PIN_FLD_EXTENDED_INFO field of the input flist.

  If the reason for terminating the call is DESTINATION_BUSY, NO_ANSWER, CALLER_ABANDON, or SMS_FAILURE, PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION is called.

- **Roaming requests**. See "About Supporting Roaming Requests" for more information.

- **Call forwarding requests**. See "About Supporting Call-Forwarding Requests" for more information.

In addition to the AAA messages, the gateway supports the following MBI administrative messages:

- **StartUp_req** - Immediately after a TCP/IP connection is established between an SEP and the gateway, the SEP sends this message to verify that the gateway is available to handle a service message. The gateway calls the PCM_OP_TCF_AAA_STOP_ACCOUNTING_OFF opcode with the service-specific (GPRS or GSM) information in the PIN_FLD_EXTENDED_INFO field to ensure that all active sessions initiated earlier by the SEP are closed.

- **StartUp_response** - After receiving a StartUp_req message, the gateway sends this message to the SEP to signal that it is ready to process a service message.

- **KeepAlive_req** - While processing an active session, the SEP sends this message to verify that the gateway is still available.

  **Note**: AAA Gateway Manager does not send KeepAlive_req to the IN.

- **KeepAlive_response** - After receiving a KeepAlive_req message, the gateway sends this message to the SEP to signal that it is still available. If the gateway does not send this message in a timely fashion, the SEP resends its KeepAlive_req message for a user-specified number of times, after which it sends a ShutDown_req message.

- **ShutDown_req** - The SEP sends this message to the gateway to signal that the SEP is shutting down. The gateway calls the PCM_OP_TCF_AAA_ACCOUNTING_OFF opcode with the service-specific (GPRS or GSM) information in the PIN_FLD_EXTENDED_INFO field to ensure that currently active sessions initiated by the SEP are closed.
ShutDown_response - After receiving a ShutDown_req message, the gateway must close all active sessions and then send this message to the SEP. The SEP then terminates its connection to the gateway.

AAA Gateway Manager also maps fields in the Cmd_ind and Cmd_res messages to EDR container fields. To process the messages, you must write a custom opcode and customize the MBI input grammar file (Pipeline_home/formatDesc/Formats/MBI/MBI_InGrammar.dsc) and MBI output grammar file (Pipeline_home/formatDesc/Formats/MBI/MBI_OutGrammar.dsc) to call your custom opcodes for those messages. See "Customizing AAA Gateway Manager Protocol Support" for more information on customizing grammar and mapping files.

About Supporting Roaming Requests

AAA Gateway Manager processes roaming requests supported in MBI v1.29 by mapping MSC_address of the mobile originating (MO) request to ORIGIN_SID and by mapping the mobile terminating (MT) request to DESTINATION_SID. AAA Gateway Manager passes these fields to the AAA opcodes.

You can use the rating configuration to create zone maps in your rate plans with different rates based on ORIGIN_SID and DESTINATION_SID. Rating opcodes determine if a call is a roaming call based on ORIGIN_SID and DESTINATION_SID and calculate the rates for roaming calls.

About Supporting Call-Forwarding Requests

AAA Gateway Manager supports MBI call-forwarding requests. Fields in the MO, MT, and mobile forwarding (MF) requests are mapped as shown in Table 2–21 to the input flist fields of the Services Framework AAA opcodes.

You can use the rating configuration to charge the subscribers different rates based on the type of the request-MO, MT, or MF.

For example, consider a scenario where a call is made from subscriber A to subscriber B and the call is forwarded to subscriber C. If all these numbers belong to prepaid accounts, AAA Gateway Manager receives an MO, MF, and MT request with the appropriate fields filled. You can create a different rate for each request type in your rating configuring and charge the accounts as follows:

- MO request: Charge A for initiating the call.
- MF request: Charge B for forwarding calls.
- MT request: Charge C for receiving calls.

If the call was made or received when the subscribers are roaming, you can use the MSC_address field in the request to apply roaming charges to the calls. See "About Supporting Roaming Requests" for more information.
You can also determine MO, MF, and MT type requests by using the PIN_FLD_DIRECTION field in the Services Framework AAA opcodes input flist. You can use the rating configuration to apply different rates based on the value in this field.

**MBI CDR Protocol Support Files**

If AAA Gateway Manager is unavailable when an HP OpenCall MBI-based network sends it an end request (End_req) message, the network saves the session data in MBI CDR file format. MBI CDRs contain information about the entire session, from start to end. When AAA Gateway Manager is available again, the MBI CDRs are processed by the gateway’s "MBI_CDR_PlaybackPipeline" pipeline. MBI CDRs are mapped to PCM_OP_TCF_AAA_STOP_ACCOUNTING for GSM and GPRS service requests.

To convert the CDRs into a format that the pipeline understands, AAA Gateway Manager uses the following files:

- MBI CDR Input Grammar
- MBI CDR Input Mapping
- MBI CDR Stream Format Description

---

**Note:** No output grammar or mapping files are provided for this protocol because AAA Gateway Manager does not need to respond to MBI CDRs.

---

**MBI CDR Input Grammar**

The MBI CDR input grammar file (Pipeline_home/formatDesc/Formats/MBI_CDR/OCSAC_CDR_v1_InGrammar.dsc) contains instructions for generating AAA EDR containers and filling them with data from MBI CDRs. At startup, AAA Gateway Manager uses this grammar to create a parser that implements the instructions.

**Error Handling**

The grammar instructs the parser to use the edrAddError function to report errors, such as a failure to create a data block.

These errors are recorded in the AAA Gateway Manager stream log (Pipeline_home/log/stream/InOutputlog_timestamp_transactionID.log). The block in which the error occurred is specified in the log.

---

**MBI CDR Input Mapping**

To transfer data from fields in MBI CDRs to the appropriate fields in AAA EDR containers, the parser follows the mapping in the flist input mapping file (Pipeline_home/formatDesc/Formats/MBI_CDR/OCSAC_CDR_v1_InMap.dsc).

---

**MBI CDR Stream Format Description**

The MBI stream format description file (Pipeline_home/formatDesc/Formats/MBI_CDR/OCSAC_CDR_v1.dsc) specifies the data type of the fields in MBI CDRs that are supported by AAA Gateway Manager.
Customizing AAA Gateway Manager Protocol Support

When AAA Gateway Manager processes network messages, the opcodes it calls, the information it passes to the opcodes, and the information it receives from the opcodes and sends back to the network are determined by the AAA EDR container description (see "AAA EDR Container Description"). By default, this description supports fields in MBI-based messages that are required to perform AAA for GSM and GPRS services.

To pass more information to and from the default opcodes, to call additional opcodes, or to support other services, you must customize the description and its associated grammar and mapping files.

To modify the AAA EDR container description, see "Modifying and Loading the EDR Container Description" in BRM Setting Up Pricing and Rating for more information.

To modify the associated grammar and mapping files, see these topics:
- Configuring EDR input processing in BRM Configuring Pipeline Rating and Discounting
- Configuring EDR output processing in BRM Configuring Pipeline Rating and Discounting

Example of Configuring the Gateway to Support Additional MBI Data

The following example shows how to configure AAA Gateway Manager to handle an additional MBI protocol field called MY_FLD. This field will store integers. It will not be processed by an opcode. Instead, it will store data from incoming MBI-based messages and enable the gateway to include the data in outgoing MBI-based messages.

1. Modify the AAA EDR container description file (aaaContainerDesc.dsc) as follows:

   Add the following line anywhere in the file’s ASSOCIATED_PROTOCOL_INFO.ASSOCIATED_MBI_INFO block:

   ```
   Integer MY_FLD;
   ```

   See "AAA EDR Container Description" for more information about this file.

2. Modify the MBI stream format description file (MBI_v1.dsc) as follows:

   Using the protocol-defined tag value of MY_FLD, add the field definition to the file. For example, if the tag value is 999, add the following definition to the file:

   ```
   MY_FLD(TLV)
   {
   Info
   {
   }
   MY_FLD TlvInteger(999, 2, 2);
   }
   ```

   See "MBI Stream Format Description" for more information about this file.

3. Modify the MBI input grammar file (MBI_v1_InGrammar.dsc) as follows:

   In the mbi_item section of the file’s Grammar block, add the lines shown here in bold:

   ```
   mbi_item:
   message_type
   | scp_name
   ```
Customizing AAA Gateway Manager Protocol Support

| protocol_version
|...
|...
| my_fld
|
...
...
my_fld:
MBI.MY_FLD
{
   //call the mapping file to map the field to the EDR
   //container
   edrInputMap("MBI.MY_FLD.STD_MAPPING")
}
;

See "MBI Input Grammar" for more information about this file.

4. Modify the MBI input mapping file (MBI_v1_InMap.dsc) as follows:

Anywhere in the MBI block of the input mapping file, specify the AAA EDR container field that should receive data from MY_FLD:

MY_FLD
{
   STD_MAPPING
   {
      MY_FLD -> DETAIL.ASS_PROTOCOL_INFO.ASS_MBI_INFO.MY_FLD;
   }
}

See "MBI Input Mapping" for more information about this file.

5. Modify the MBI output grammar file (MBI_v1_OutGrammar.dsc) as follows:

In the section of the file where the response in which this field must be included is constructed, add all of the following code:

   //Check whether the associated EDR field contains a value:
   if(edrInternalState(DETAIL.ASS_PROTOCOL_INFO.ASS_MBI_INFO.
   BALANCE_TYPE, 0, 0)!= CLEARED)
   {
      //If it does, write the value to the buffer:
      edrOutputMapToBuffer(optMbiBuffer, "MBI.BALANCE_TYPE.STD_ MAPPING", 0, 0);
      }
   ...
   ...
   //After all fields are written to the buffer, write the buffer
   //to the response:
   outputWrite(headerBuffer + optMbiBuffer);

See "MBI Output Grammar" for more information about this file.

6. Modify the MBI output mapping file (MBI_v1_OutMap.dsc) as follows:

Anywhere in the MBI block of the output mapping file, add the following code:

MY_FLD
{
   STD_MAPPING
   {
      //...
Example of Configuring the Gateway to Support Additional Diameter Data

The following example shows how to configure AAA Gateway Manager to handle an additional Diameter protocol field called MY_FLD. This field will store integers. It will not be processed by an opcode. Instead, it will store data from incoming Diameter-based messages and enable the gateway to include the data in outgoing Diameter-based messages.

1. Modify the AAA EDR container description file (aaaContainerDesc.dsc) as follows:

Add the following line anywhere in the file's ASSOCIATED_PROTOCOL_INFO. ASSOCIATED_DIAMETER_INFO block:

```
Integer MY_FLD;
```

See "AAA EDR Container Description" for more information about this file.

2. Modify the Diameter stream format description file (Diameter_v01.dsc) as follows:

Using the protocol-defined tag value of MY_FLD, add the field definition to the file. For example, if the tag value is 999, add the following definition to the file:

```
MY_FLD(TLV)
{
  Info
  {
    MY_FLD DiameterInteger(999, "M", 0, 32);
  }
}
```

See "Diameter Stream Format Description" for more information about this file.

3. Modify the Diameter input grammar file (Diameter_v01_InGrammar.dsc) as follows:

In the `diameterItem` section of the file's Grammar block, add the lines shown here in bold:

```
diameterItem:
  cc_request_type
  | sessionid
  | accounting_record_type
  ... 
  ... 
  | my_fld
  ;
...
...
my_fld:
  DIAMETER.MY_FLD
  {
    //call the mapping file to map the field to the EDR
    //container
    edrInputMap("DIAMETER.MY_FLD. STD_MAPPING ");
  }
  ;
```
4. Modify the Diameter input mapping file (Diameter_v01_InMap.dsc) as follows:

   Anywhere in the Diameter block of the input mapping file, specify the AAA EDR container field that should receive data from MY_FLD:

   ```
   MY_FLD
   {
       STD_MAPPING
       {
           MY_FLD -> DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_FLD;
       }
   }
   ```

   See "Diameter Input Mapping" for more information about this file.

5. Modify the Diameter output grammar file (Diameter_v01_OutGrammar.dsc) as follows:

   In the section of the file where the response in which this field must be included is constructed, add all of the following code:

   ```
   //Check whether the associated EDR field contains a value:
   if(edrInternalState(DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_FLD, 0, 0)!=
       CLEARED)
   {
       //If it does, write the value to the buffer:
       edrOutputMapToBuffer(optDiameterBuffer, "DIAMETER.MY_FLD. STD_MAPPING ",
                           0, 0);
   }
   ```

   See "Diameter Output Grammar" for more information about this file.

6. Modify the Diameter output mapping file (Diameter_v01_OutMap.dsc) as follows:

   Anywhere in the Diameter block of the output mapping file, add the following code:

   ```
   MY_FLD
   {
       STD_MAPPING
       {
           MY_FLD <- DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_FLD;
       }
   }
   ```

Example of Configuring the Gateway to Support Additional Diameter Data as Group

The following example shows how to configure AAA Gateway Manager to handle an additional Diameter protocol group called MY_GROUP which has three fields: MY_FLD1, MY_FLD2 and MY_FLD3. All fields will store integer value. It will not be processed by an opcode. Instead, it will store data from incoming Diameter-based messages and enable the gateway to include the data in outgoing Diameter-based messages.

1. Modify the AAA EDR container description file (aaaContainerDesc.dsc) as follows:
Add the following line anywhere in the file's ASSOCIATED_PROTOCOL_INFO. ASSOCIATED_DIAMETER_INFO block:

```
MY_GROUP MY_GROUP;
```

Add the following lines after that block ends:

```
MY_GROUP
{
   Integer MY_FLD1;
   Integer MY_FLD2;
   Integer MY_FLD3;
   Integer LENGTH;
}
```

See "AAA EDR Container Description" for more information about this file.

2. Modify the Diameter stream format description file (Diameter_v01.dsc) as follows:

Using the protocol-defined tag value of MY_GROUP and its fields, add the group and field definition to the file.

For example, if the tag value of MY_GROUP is 996, add the following definition to the file:

```
MY_GROUP(TLV)
{
   Info
   {
       EndRecord = BLOCK_END;
   }
   LENGTH DiameterGrouped(996, "M", 0, 64);
}
```

```
MY_FLD1(TLV)
{
   Info
   {
   }
   MY_FLD1 DiameterInteger(997, "M", 0, 32);
}
```

```
MY_FLD2(TLV)
{
   Info
   {
   }
   MY_FLD2 DiameterInteger(998, "M", 0, 32);
}
```

```
MY_FLD3(TLV)
{
   Info
   {
   }
   MY_FLD3 DiameterInteger(999, "M", 0, 32);
}
```

See "Diameter Stream Format Description" for more information about this file.

3. Modify the Diameter input grammar file (Diameter_v01_InGrammar.dsc) as follows:

```
```
Customizing AAA Gateway Manager Protocol Support

Add the function below in the iScript section of the grammar:

```plaintext
def addMyGroup
{
    if (edrAddDatablock( DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP )
== false )
    {
        errorMsg = "failed to add MY_GROUP datablock for EDR ";
        edrAddError("ERR_DIAMETER_INPUT_ERROR", 4, errorMsg);
    }
}
```

In the `diameterItem` section of the file's Grammar block, add the lines shown here in bold:

```plaintext
diameterItem:
    cc_request_type
    sessionid
    accounting_record_type
    ...
    ...
    | myGroup
    ;
    ...
    ...

myGroup:
    DIAMETER.MY_GROUP
    {
        addMyGroup();
    }

myGroupItems
    DIAMETER.BLOCK_END
    ;

myGroupItems:
    myGroupItems
    myGroupItem
    |
    ;

myGroupItem:
    my_fld1
    | my_fld2
    | my_fld3
    ;

my_fld1:
    DIAMETER.MY_FLD1
    {
        //call the mapping file to map the field to the EDR
        //container
        edrInputMap("DIAMETER.MY_FLD1.MY_GROUP");
    }
    ;

my_fld2:
    DIAMETER.MY_FLD2
    {
        //call the mapping file to map the field to the EDR
```
Customizing AAA Gateway Manager Protocol Support

AAA Gateway Manager Protocol Support

2-35

//container
edrInputMap("DIAMETER.MY_FLD2.MY_GROUP");
}
;

my_fld3:
DIAMETER.MY_FLD3
{
    //call the mapping file to map the field to the EDR
    //container
    edrInputMap("DIAMETER.MY_FLD3.MY_GROUP");
}
;

See "Diameter Input Grammar" for more information about this file.

4. Modify the Diameter input mapping file (Diameter_v01_InMap.dsc) as follows:
   Anywhere in the Diameter block of the input mapping file, specify the AAA EDR container field that should receive data from MY_GROUP:

   MY_FLD1
   {
       MY_GROUP
       {
        MY_FLD1   ->   DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.MY_FLD1;
       }  
    }

   MY_FLD2
   {
       MY_GROUP
       {
        MY_FLD2   ->   DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.MY_FLD2;
       }  
    }

   MY_FLD3
   {
       MY_GROUP
       {
        MY_FLD3   ->   DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.MY_FLD3;
       }  
    }

   See "Diameter Input Mapping" for more information about this file.

5. Modify the Diameter output grammar file (Diameter_v01_OutGrammar.dsc) as follows:
   In the section of the file where the response in which this group must be included is constructed, add all of the following code:

   // Check whether the block is empty
   Long i = 0;
   for ( i=0; i < edrNumDatablocks(DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP,0,0); i=i+1 )
   {
     String myGroupBuffer = "";

//Check whether the associated EDR field contains a value:
if(edrInternalState(DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.MY_FLD1, 0, 0, i)!= CLEARED)
{
   //If it does, write the value to the buffer:
edrOutputMapToBuffer(myGroupBuffer, "DIAMETER.MY_FLD1.MY_GROUP", 0, 0, i);
}

//Check whether the associated EDR field contains a value:
if(edrInternalState(DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.MY_FLD2, 0, 0, i)!= CLEARED)
{
   //If it does, write the value to the buffer:
edrOutputMapToBuffer(myGroupBuffer, "DIAMETER.MY_FLD2.MY_GROUP", 0, 0, i);
}

//Check whether the associated EDR field contains a value:
if(edrInternalState(DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.MY_FLD3, 0, 0, i)!= CLEARED)
{
   //If it does, write the value to the buffer:
edrOutputMapToBuffer(myGroupBuffer, "DIAMETER.MY_FLD3.MY_GROUP", 0, 0, i);
}
edrLong(DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.LENGTH, 0, 0, i) =
length(myGroupBuffer) + AVP_HEADER_LENGTH;
edrOutputMapToBuffer(optDiameterBuffer,"DIAMETER.MY_GROUP.CREDIT_CONTROL",0,0,i);
optDiameterBuffer = optDiameterBuffer + myGroupBuffer;

See "Diameter Output Grammar" for more information about this file.

6. Modify the Diameter output mapping file (Diameter_v01_OutMap.dsc) as follows:

   Anywhere in the Diameter block of the output mapping file, add the following code:

   MY_GROUP
   {
      CREDIT_CONTROL
      {
         LENGTH <- DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.LENGTH;
      }
   }

   MY_FLD1
   {
      MY_GROUP
      {
         MY_FLD1   <-   DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.MY_FLD1;
      }
   }

   MY_FLD2
   {
      MY_GROUP
      {
Example of Configuring the Gateway to Support Additional Diameter Data as Multiple Group

The following example shows how to configure AAA Gateway Manager to handle an additional Diameter protocol group called MY_GROUP which can come multiple number of times and has three fields: MY_FLD1, MY_FLD2 and MY_FLD3. All fields will store integer value. It will not be processed by an opcode. Instead, it will store data from incoming Diameter-based messages and enable the gateway to include the data in outgoing Diameter-based messages.

1. Modify the AAA EDR container description file (aaaContainerDesc.dsc) as follows:
   Add the following line anywhere in the file's ASSOCIATED_PROTOCOL_INFO.
   ASSOCIATED_DIAMETER_INFO block:
   ```
   MY_Group MY_Group;
   ```
   Add the following lines after that block ends
   ```
   MY_Group
   {
      Integer MY_FLD1;
      Integer MY_FLD2;
      Integer MY_FLD3;
      Integer LENGTH;
   }
   ```
   See "AAA EDR Container Description" for more information about this file.

2. Modify the Diameter stream format description file (Diameter_v01.dsc) as follows:
   Using the protocol-defined tag value of MY_GROUP and its fields, add the group and field definition to the file.
   For example, if the tag value of MY_GROUP is 996, add the following definition to the file:
   ```
   MY_Group(TLV)
   {
      Info
      {
         EndRecord = BLOCK_END;
      }
      LENGTH DiameterGrouped(996, "M", 0 , 64);
   }
   ```
MY_FLD1(TLV)
{
    Info
    {
    }
    MY_FLD1 DiameterInteger(997, "M", 0, 32);
}

MY_FLD2(TLV)
{
    Info
    {
    }
    MY_FLD2 DiameterInteger(998, "M", 0, 32);
}

MY_FLD3(TLV)
{
    Info
    {
    }
    MY_FLD3 DiameterInteger(999, "M", 0, 32);
}

See “Diameter Stream Format Description” for more information about this file.

3. Modify the Diameter input grammar file (Diameter_v01_InGrammar.dsc) as follows:

Add the below function in the iScript section of the grammar:

    Long index = 0;
    function addMyGroup
    {
        if (edrAddDatablock( DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP )
            == false )
        {
            errorMsg = "failed to add MY_GROUP datablock for EDR ";
            edrAddError( "ERR_DIAMETER_INPUT_ERROR", 4, errorMsg);
        }
        else
        {
            edrArrayIndex( DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP,
                0, 0, 0) = index;
            index = index + 1;
        }
    }

In the diameterItem section of the file’s Grammar block, add the lines shown here in bold:

diameterItem:
cc_request_type
| sessionid
| accounting_record_type
| ...
| myGroup
| ...
| ...

myGroup:
DIAMETER.MY_GROUP
{
    addMyGroup();
}

myGroupItems
DIAMETER.BLOCK_END
;

myGroupItems:
    myGroupItems
    myGroupItem
    |
;

myGroupItem:
    my_fld1
    | my_fld2
    | my_fld3
;

my_fld1:
DIAMETER.MY_FLD1
{
    //call the mapping file to map the field to the EDR
    //container
    edrInputMap("DIAMETER.MY_FLD1.MY_GROUP");
}
;

my_fld2:
DIAMETER.MY_FLD2
{
    //call the mapping file to map the field to the EDR
    //container
    edrInputMap("DIAMETER.MY_FLD2.MY_GROUP");
}
;

my_fld3:
DIAMETER.MY_FLD3
{
    //call the mapping file to map the field to the EDR
    //container
    edrInputMap("DIAMETER.MY_FLD3.MY_GROUP");
}
;

See "Diameter Input Grammar" for more information about this file.

4. Modify the Diameter input mapping file (Diameter_v01_InMap.dsc) as follows:
   Anywhere in the Diameter block of the input mapping file, specify the AAA EDR container field that should receive data from MY_GROUP:

   MY_FLD1
   {
       MY_GROUP
       {
           MY_FLD1   ->   DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_
GROUP.MY_FLD1;
}

MY_FLD2
{
    MY_GROUP
    {
        MY_FLD2 -> DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_
GROUP.MY_FLD2;
    }
}

MY_FLD3
{
    MY_GROUP
    {
        MY_FLD3 -> DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_
GROUP.MY_FLD3;
    }
}

See "Diameter Input Mapping" for more information about this file.

5. Modify the Diameter output grammar file (Diameter_v01_OutGrammar.dsc) as follows:

In the section of the file where the response in which this group must be included is constructed, add all of the following code:

```java
// Check whether the block is empty
Long i = 0;
for ( i=0; i < edrNumDatablocks(DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_
GROUP,0,0); i=i+1 )
{
    String myGroupBuffer = "";

    //Check whether the associated EDR field contains a value:
    if(edrInternalState(DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_
GROUP.MY_FLD1, 0, 0, i)!= CLEARED)
    {
        //If it does, write the value to the buffer:
        edrOutputMapToBuffer(myGroupBuffer, "DIAMETER.MY_FLD1.MY_GROUP", 0,
0, i);
    }

    //Check whether the associated EDR field contains a value:
    if(edrInternalState(DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_
GROUP.MY_FLD2, 0, 0, i)!= CLEARED)
    {
        //If it does, write the value to the buffer:
        edrOutputMapToBuffer(myGroupBuffer, "DIAMETER.MY_FLD2.MY_GROUP", 0,
0, i);
    }

    //Check whether the associated EDR field contains a value:
    if(edrInternalState(DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_
GROUP.MY_FLD3, 0, 0, i)!= CLEARED)
```
Configuring AAA Gateway Manager to Accept AVPs at Any Level

To define any AVP at any level, you must add an end of block marker to your Diameter stream format description and input grammar files:

```
{               //If it does, write the value to the buffer:
edrOutputMapToBuffer(myGroupBuffer, "DIAMETER.MY_FLD3.MY_GROUP", 0, 0, i);
}
edrLong(DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.LENGTH, 0,0,i) =
      length(myGroupBuffer) + AVP_HEADER_LENGTH;
edrOutputMapToBuffer(optDiameterBuffer,"DIAMETER.MY_GROUP.CREDIT_CONTROL",0,0,i);
      optDiameterBuffer = optDiameterBuffer + myGroupBuffer;
}

See "Diameter Output Grammar" for more information about this file.

6. Modify the Diameter output mapping file (Diameter_v01_OutMap.dsc) as follows:

Anywhere in the Diameter block of the output mapping file, add the following code:

```
MY_GROUP
{
    CREDIT_CONTROL
    {
        LENGTH <- DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.LENGTH;
    }
}
MY_FLD1
{
    MY_GROUP
    {
        MY_FLD1   <-   DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.MY_FLD1;
    }
}
MY_FLD2
{
    MY_GROUP
    {
        MY_FLD2   <-   DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.MY_FLD2;
    }
}
MY_FLD3
{
    MY_GROUP
    {
        MY_FLD3   <-   DETAIL.ASS_PROTOCOL_INFO.ASS_DIAMETER_INFO.MY_GROUP.MY_FLD3;
    }
}
```

Configuring AAA Gateway Manager to Accept AVPs at Any Level
1. Open the Diameter stream format description file (Pipeline_home/formatDesc/Formats/Diameter/Diameter_v01.dsc) in a text editor.

2. Define a special end of block marker by adding the following lines to the file:

```plaintext
BLOCK_END(FIX)
{
  Info
  {
    Pattern = "\0\0\0\0";
  }
  BLOCK_END AscString(4);
}
```

3. Add a new end of block marker to all Grouped AVP definitions in the file.

   For example:

   ```plaintext
   CC_MONEY(TLV)
   {
     Info
     {
       EndRecord = BLOCK_END; // This is the new line added
     }
     LENGTH DiameterGrouped(413, "M", 0);
   }
   ```

4. Save and close the file.

5. Open the Diameter input grammar file (Pipeline_home/formatDesc/Formats/Diameter/Diameter_v01_InGrammar.dsc) in a text editor.

6. Add a DIAMETER.BLOCK_END line at the end of each Grouped AVP.

   For example:

   ```plaintext
   reqCCmoney:
     DIAMETER.CC_MONEY
     {
       addRequestedServiceUnitCCMoneyDataBlock();
     }
   reqCCMoneyItems
   DIAMETER.BLOCK_END
   ;
   ```

   ```plaintext
   reqCCMoneyItems:
   reqCCMoneyItems
   reqCCMoneyItem
   | /* EMPTY */
   ;
   ```

   ```plaintext
   reqCCMoneyItem:
     DIAMETER.CURRENCY_CODE
     {
       logStdout("CurrencyCode: " +
       longToStr(tokenLong("DIAMETER.CURRENCY_CODE.CURRENCY_CODE")) + '\n";
       mapreqCurrencyCode();
     }
     | DIAMETER.UNKNOWN_DATA
     {
       logStdout("Unknown Data Inside DIAMETER.CC_MONEY\n"});
     }
   ```
7. Save and close the file.
This document provides an overview of the event data record (EDR) container
description for Oracle Communications Billing and Revenue Management (BRM)
AAA Gateway Manager.

**Note:** The AAA EDR container description is different from the rating
EDR container description. See "BRM Rating EDR Container
Description" in *BRM Configuring Pipeline Rating and Discounting* for
information about the latter.

See "Using BRM AAA Gateway Manager" for information about AAA Gateway
Manager.

See "AAA Gateway Manager Protocol Support" for information about the grammar
and mapping files used to translate network requests into AAA EDR containers and
vice versa.

See "About EDRs" in *BRM Configuring Pipeline Rating and Discounting* for general
information about EDR container descriptions.

### About the AAA EDR Container Description

To process a request from a prepaid network, AAA Gateway Manager must convert
the request into a format that its pipeline modules understand. This format is defined
in the AAA EDR container description file (`aaaContainerDesc.dsc`) in the `Pipeline_
home/formatDesc/Formats/Portal` directory.

The AAA EDR container description includes the following EDR content types:
- HEADER Content Type for AAA EDRs
- DETAIL Content Type for AAA EDRs

It also includes a block of fields in its INTERNAL record that appears in all SERVICE
type AAA EDR containers. See "INTERNAL Record in AAA EDRs" for more
information.

### HEADER Content Type for AAA EDRs

The HEADER content type for AAA EDRs is defined by the HEADER record in the
AAA EDR container description. The information in this record is used by the OUT_
Serialize module to write a serialized representation of an AAA EDR to a file in binary
format.

The AAA EDR HEADER record contains the fields shown in Table 3–1.
DETAIL Content Type for AAA EDRs

The DETAIL content type for AAA EDRs is defined by the DETAIL record in the AAA EDR container description. The information in this record is used to perform AAA activities.

The AAA EDR DETAIL record contains these fields and blocks:

- **Function Module Fields**
  - ASSOCIATED_SCP_REQUEST Block
  - ASSOCIATED_OPERATIONAL_REQUEST Block
  - ASSOCIATED_TCF_AAA_DETAIL block
  - PCM_OP_TCF_AAA_AUTHORIZE Block
  - PCM_OP_TCF_AAA_UPDATE_ACCOUNTING Block
  - PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION Block
  - PCM_OP_TCF_AAA_START_ACCOUNTING Block
  - PCM_OP_TCF_AAA_STOP_ACCOUNTING Block
  - PCM_OP_TCF_AAA_QUERY_BALANCE Block
  - PCM_OP_TCF_AAA_REFUND Block
  - PCM_OP_TCF_AAA_SERVICE_PRICE_ENQUIRY Block
  - PCM_OP_TCF_AAA_ACCOUNTING_ON Block
  - PCM_OP_TCF_AAA_ACCOUNTING_OFF Block
  - ASSOCIATED_PROTOCOL_INFO Block
  - PCM_OP_EBUF Block

**Function Module Fields**

These fields shown in Table 3–2 are required by the following function modules: FCT_Timer, FCT_CancelTimer, FCT_Opcode, FCT_Zone, and FCT_NumberPortability.

---

**Table 3–1 AAA EDR Header Record Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATION_PROCESS</td>
<td>String</td>
<td>Specifies the process that created the input file. Can be set to one of these values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ BACKOUT_PIPELINE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ EVENT_EXTRACTION_TOOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ PIN_TRANSFORM_CDR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RATING_PIPELINE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RERATING_PIPELINE</td>
</tr>
<tr>
<td>EVENT_TYPE</td>
<td>String</td>
<td>Must be present for the OUT_Serialize module to process the AAA EDR container. No value, however, is put in this field.</td>
</tr>
</tbody>
</table>

---
### Table 3–2  Required Function Module Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPCODE_NODE</td>
<td>String</td>
<td>EDR container node that specifies the fields to map to the input flist of the opcode specified in the OPCODE_NUM field. For example, if OPCODE_NUM specifies PCM_OP_GSM_AAA_AUTHORIZE, the value of OPCODE_NODE must be DETAIL.ASS_GSM_AAA_DETAIL.PCM_OP_GSM_AAA_AUTHORIZE. Required by FCT_Opcode.</td>
</tr>
<tr>
<td>OPCODE_NUM</td>
<td>Integer</td>
<td>Number of the BRM opcode that performs the requested action. Opcode numbers are defined in header (*.h) files in the BRM_home/include/ops directory. Required by FCT_Opcode.</td>
</tr>
<tr>
<td>OPCODE_FLAG</td>
<td>Integer</td>
<td>Flag to include in the call to the opcode that performs the requested action. Required by FCT_Opcode.</td>
</tr>
<tr>
<td>SOCKET_ID</td>
<td>Integer</td>
<td>Number of the prepaid network socket to which AAA Gateway Manager sends its response. Required by FCT_Opcode.</td>
</tr>
<tr>
<td>CURRENT_TIME</td>
<td>Integer</td>
<td>Time when the EDR container is created. Required by FCT_Timer.</td>
</tr>
<tr>
<td>TIMEOUT_OFFSET</td>
<td>Integer</td>
<td>Number of seconds after the EDR is created that it times out (expires). Required by FCT_Timer.</td>
</tr>
<tr>
<td>TIMER_ID</td>
<td>Integer</td>
<td>ID assigned to the EDR container’s timer when FCT_SetTimer schedules it. This ID is required to cancel the timer. Required by FCT_Timer.</td>
</tr>
<tr>
<td>TIMEOUT_FLAG</td>
<td>Integer</td>
<td>Specifies if the EDR has timed out. Required by FCT_Timer.</td>
</tr>
<tr>
<td>SESSION_ID</td>
<td>String</td>
<td>Timer id needed to cancel the timer Required by FCT_Timer.</td>
</tr>
<tr>
<td>REACTOR_IDPRE</td>
<td>Integer</td>
<td>Internal Field: 1st part of the reactor’s address</td>
</tr>
<tr>
<td>REACTOR_IDPOST</td>
<td>Integer</td>
<td>Internal Field: 2nd part of the reactor’s address</td>
</tr>
<tr>
<td>MILLISEC_TIME</td>
<td>Integer</td>
<td>Specifies the latency time in milliseconds. Required by FCT_Timer.</td>
</tr>
<tr>
<td>CHARGING_START_TIMESTAMP</td>
<td>Date</td>
<td>Start time of chargeable service. Required by FCT_Zone.</td>
</tr>
<tr>
<td>RETAIL_IMPACT_CATEGORY</td>
<td>String</td>
<td>Impact category that determines the rate of the usage scenario, such as the zone value used for customer rating. Required by FCT_Zone.</td>
</tr>
<tr>
<td>WHOLESALE_IMPACT_CATEGORY</td>
<td>String</td>
<td>Impact category used for rating purchases. Required by FCT_Zone.</td>
</tr>
<tr>
<td>ZONE_ENTRY_NAME</td>
<td>String</td>
<td>Description of call destination to display on invoices. Required by FCT_Zone.</td>
</tr>
</tbody>
</table>
### Table 3–2  (Cont.) Required Function Module Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZONE_DESCRIPTION</td>
<td>String</td>
<td>Description of call zone to display on invoices. Required by FCT_Zone.</td>
</tr>
<tr>
<td>INTERN_SERVICE_CODE</td>
<td>String</td>
<td>Internal service code that determines which EDR container fields are used to find the customer account. For example, a telephone service uses the A (originating) telephone number to find the customer account. Required by FCT_Zone.</td>
</tr>
<tr>
<td>INTERN_A_NUMBER_CODE</td>
<td>String</td>
<td>A (originating) telephone number. Required by FCT_Zone.</td>
</tr>
<tr>
<td>INTERN_B_NUMBER_CODE</td>
<td>String</td>
<td>B (destination) telephone number. Required by FCT_Zone.</td>
</tr>
<tr>
<td>INTERN_ZONE_MODEL</td>
<td>Integer</td>
<td>Zone model ID associated with the prepaid service. Required by FCT_Zone.</td>
</tr>
<tr>
<td>INTERN_APN_GROUP</td>
<td>String</td>
<td>Zone model–related APN_GROUP used by the FCT_APN_Map module. Required by FCT_Zone.</td>
</tr>
<tr>
<td>EVENT_TYPE</td>
<td>String</td>
<td>Event type associated with the prepaid service. Required by FCT_Zone.</td>
</tr>
<tr>
<td>ASS_SCP_REQUEST</td>
<td>ASSOCIATED_SCP_REQUEST</td>
<td>Block containing information about the prepaid network request. See &quot;ASSOCIATED_SCP_REQUEST Block&quot; for more information.</td>
</tr>
<tr>
<td>ASS_OPERATIONAL_REQUEST</td>
<td>ASSOCIATED_OPERATIONAL_REQUEST</td>
<td>Block containing network-specific operational messages. See &quot;ASSOCIATED_OPERATIONAL_REQUEST Block&quot; for more information.</td>
</tr>
<tr>
<td>ASS_TCF_AAA_DETAIL</td>
<td>ASSOCIATED_TCF_AAA_DETAIL</td>
<td>Block containing opcodes required to implement TCF services. See &quot;ASSOCIATED_TCF_AAA_DETAIL block&quot; for more information.</td>
</tr>
<tr>
<td>PCM_OP_EBUF</td>
<td>PCM_OP_EBUF</td>
<td>Block containing opcode error information. See &quot;PCM_OP_EBUF Block&quot; for more information.</td>
</tr>
<tr>
<td>ASS_PROTOCOL_INFO</td>
<td>ASSOCIATED_PROTOCOL_INFO</td>
<td>Block containing protocol information. See &quot;ASSOCIATED_PROTOCOL_INFO Block&quot; for more information.</td>
</tr>
<tr>
<td>A_NUMBER</td>
<td>String</td>
<td>Specifies the event originator. Used for number portability.</td>
</tr>
<tr>
<td>B_NUMBER</td>
<td>String</td>
<td>Specifies the event receiver. Used for number portability.</td>
</tr>
<tr>
<td>CHARGING_START_TIMESTAMP</td>
<td>Date</td>
<td>Specifies the event charging timestamp. Used for number portability.</td>
</tr>
<tr>
<td>SOURCE_NETWORK</td>
<td>String</td>
<td>Specifies the source network. This can either be the PLMN ID or any logical operator code. Used for number portability.</td>
</tr>
</tbody>
</table>
ASSOCIATED_SCP_REQUEST Block
This block contains the fields shown in Table 3–3.

<table>
<thead>
<tr>
<th>Field</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOURCE_NETWORK_TYPE</td>
<td>String</td>
<td>Optional. Specifies the source network type, for example GSM 900. Used for number portability.</td>
</tr>
<tr>
<td>DESTINATION_NETWORK</td>
<td>String</td>
<td>Specifies the network to which an event is routed. Used for number portability.</td>
</tr>
<tr>
<td>DESTINATION_NETWORK_TYPE</td>
<td>String</td>
<td>Optional. Specifies the destination network type, for example GSM 900. Used for number portability.</td>
</tr>
<tr>
<td>IGNORE_NP</td>
<td>Integer</td>
<td>State variable to indicate NP data needs to be looked up. Used for number portability.</td>
</tr>
</tbody>
</table>

ASSOCIATED_OPERATIONAL_REQUEST Block
This block can contain zero or more fields for network-specific operational messages supported by AAA Gateway Manager.

By default, this block is empty. To add fields to it, see "Customizing the AAA EDR Container Description" for more information.

ASSOCIATED_TCF_AAA_DETAIL block
This block includes blocks for every opcode required to implement AAA for GSM services as shown in Table 3–4.
Table 3–4  ASSOCIATED_TCF_AAA_DETAIL Block Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCM_OP_TCF_AAA_AUTHENTICATE</td>
<td>Block that contains the input and output flist fields for the PCM_OP_TCF_AAA_AUTHENTICATE opcode. See &quot;PCM_OP_TCF_AAA_AUTHENTICATE Block&quot; for more details.</td>
</tr>
<tr>
<td>PCM_OP_TCF_AAA_AUTHORIZE</td>
<td>Block that contains the input and output flist fields for the PCM_OP_TCF_AAA_AUTHORIZE opcode. See &quot;PCM_OP_TCF_AAA_AUTHORIZE Block&quot; for more details.</td>
</tr>
<tr>
<td>PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE</td>
<td>Block that contains the input and output flist fields for the PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode. See &quot;PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE Block&quot; for more details.</td>
</tr>
<tr>
<td>PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION</td>
<td>Block that contains the input and output flist fields for the PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION opcode. See &quot;PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION Block&quot; for more details.</td>
</tr>
<tr>
<td>PCM_OP_TCF_AAA_START_ACCOUNTING</td>
<td>Block that contains the input and output flist fields for the PCM_OP_TCF_AAA_START_ACCOUNTING opcode. See &quot;PCM_OP_TCF_AAA_START_ACCOUNTING Block&quot; for more details.</td>
</tr>
<tr>
<td>PCM_OP_TCF_AAA_UPDATE_ACCOUNTING</td>
<td>Block that contains the input and output flist fields for the PCM_OP_TCF_AAA_UPDATE_ACCOUNTING opcode. See &quot;PCM_OP_TCF_AAA_UPDATE_ACCOUNTING Block&quot; for more details.</td>
</tr>
<tr>
<td>PCM_OP_TCF_AAA_STOP_ACCOUNTING</td>
<td>Block that contains the input and output flist fields for the PCM_OP_TCF_AAA_STOP_ACCOUNTING opcode. See &quot;PCM_OP_TCF_AAA_STOP_ACCOUNTING Block&quot; for more details.</td>
</tr>
<tr>
<td>PCM_OP_TCF_AAA_QUERY_BALANCE</td>
<td>Block that contains the input and output flist fields for the PCM_OP_TCF_AAA_QUERY_BALANCE opcode. See &quot;PCM_OP_TCF_AAA_QUERY_BALANCE Block&quot; for more details.</td>
</tr>
<tr>
<td>PCM_OP_TCF_AAA_SERVICE_PRICE_ENQUIRY</td>
<td>Block that contains the input and output flist fields for the PCM_OP_TCF_AAA_SERVICE_PRICE_ENQUIRY opcode. See &quot;PCM_OP_TCF_AAA_SERVICE_PRICE_ENQUIRY Block&quot; for more details.</td>
</tr>
<tr>
<td>PCM_OP_TCF_AAA_ACCOUNTING_ON</td>
<td>Block that contains the input and output flist fields for the PCM_OP_TCF_AAA_ACCOUNTING_ON opcode. See &quot;PCM_OP_TCF_AAA_ACCOUNTING_ON Block&quot; for more details.</td>
</tr>
<tr>
<td>PCM_OP_TCF_AAA_ACCOUNTING_OFF</td>
<td>Block that contains the input and output flist fields for the PCM_OP_TCF_AAA_ACCOUNTING_OFF opcode. See &quot;PCM_OP_TCF_AAA_ACCOUNTING_OFF Block&quot; for more details.</td>
</tr>
</tbody>
</table>

**PCM_OP_TCF_AAA_AUTHENTICATE Block**

This block contains the fields shown in Table 3–5.
For information about fields and blocks in the INPUT and OUTPUT blocks, see the `Pipeline_home/formatDesc/Formats/Portal/aaaContainerDesc.dsc` file.

**PCM_OP_TCF_AAA_AUTHORIZE Block**

This block contains the fields shown in Table 3–6.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>Block that contains the input flist fields for this opcode.</td>
</tr>
<tr>
<td></td>
<td>Data type is <code>PCM_OP_TCF_AAA_AUTHORIZE_INPUT</code>.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Block that contains the output flist fields for this opcode.</td>
</tr>
<tr>
<td></td>
<td>Data type is <code>PCM_OP_TCF_AAA_AUTHORIZE_OUTPUT</code>.</td>
</tr>
</tbody>
</table>

For information about fields and blocks in the INPUT and OUTPUT blocks, see the `Pipeline_home/formatDesc/Formats/Portal/aaaContainerDesc.dsc` file.

**PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE Block**

This block contains the fields shown in Table 3–7.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>Block that contains the input flist fields for this opcode.</td>
</tr>
<tr>
<td></td>
<td>Data type is <code>PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE_INPUT</code>.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Block that contains the output flist fields for this opcode.</td>
</tr>
<tr>
<td></td>
<td>Data type is <code>PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE_OUTPUT</code>.</td>
</tr>
</tbody>
</table>

For information about fields and blocks in the INPUT and OUTPUT blocks, see the `Pipeline_home/formatDesc/Formats/Portal/aaaContainerDesc.dsc` file.

**PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION Block**

This block contains the fields shown in Table 3–8.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>Block that contains the input flist fields for this opcode.</td>
</tr>
<tr>
<td></td>
<td>Data type is <code>PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION_INPUT</code>.</td>
</tr>
</tbody>
</table>
About the AAA EDR Container Description

**PCM_OP_TCF_AAA_START_ACCOUNTING Block**
This block contains the fields shown in Table 3–9.

**Table 3–9 PCM_OP_TCF_AAA_START_ACCOUNTING Block Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>Block that contains the input flist fields for this opcode. Data type is PCM_OP_TCF_AAA_START_ACCOUNTING_INPUT.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Block that contains the output flist fields for this opcode. Data type is PCM_OP_TCF_AAA_START_ACCOUNTING_OUTPUT.</td>
</tr>
</tbody>
</table>

For information about fields and blocks in the INPUT and OUTPUT blocks, see the Pipeline_home/formatDesc/Formats/Portal/aaaContainerDesc.dsc file.

**PCM_OP_TCF_AAA_UPDATE_ACCOUNTING Block**
This block contains the fields shown in Table 3–10.

**Table 3–10 PCM_OP_TCF_AAA_UPDATE_ACCOUNTING Block Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>Block that contains the input flist fields for this opcode. Data type is PCM_OP_TCF_AAA_UPDATE_ACCOUNTING_INPUT.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Block that contains the output flist fields for this opcode. Data type is PCM_OP_TCF_AAA_UPDATE_ACCOUNTING_OUTPUT.</td>
</tr>
</tbody>
</table>

For information about fields and blocks in the INPUT and OUTPUT blocks, see the Pipeline_home/formatDesc/Formats/Portal/aaaContainerDesc.dsc file.

**PCM_OP_TCF_AAA_STOP_ACCOUNTING Block**
This block contains the fields shown in Table 3–11.

**Table 3–11 PCM_OP_TCF_AAA_STOP_ACCOUNTING Block Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>Block that contains the input flist fields for this opcode. Data type is PCM_OP_TCF_AAA_STOP_ACCOUNTING_INPUT.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Block that contains the output flist fields for this opcode. Data type is PCM_OP_TCF_AAA_STOP_ACCOUNTING_OUTPUT.</td>
</tr>
</tbody>
</table>
For information about fields and blocks in the INPUT and OUTPUT blocks, see the Pipeline_home/formatDesc/Formats/Portal/aaaContainerDesc.dsc file.

**PCM_OP_TCF_AAA_QUERY_BALANCE Block**
This block contains the fields shown in Table 3–12.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| INPUT   | Block that contains the input flist fields for this opcode.  
          | Data type is PCM_OP_TCF_AAA_QUERY_BALANCE_INPUT.                                  |
| OUTPUT  | Block that contains the output flist fields for this opcode.  
          | Data type is PCM_OP_TCF_AAA_QUERY_BALANCE_OUTPUT.                                 |

**PCM_OP_TCF_AAA_SERVICE_PRICE_ENQUIRY Block**
This block contains the fields shown in Table 3–13.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| INPUT   | Block that contains the input flist fields for this opcode.  
          | Data type is PCM_OP_TCF_AAA_SERVICE_PRICE_ENQUIRY_INPUT.                                  |
| OUTPUT  | Block that contains the output flist fields for this opcode.  
          | Data type is PCM_OP_TCF_AAA_SERVICE_PRICE_ENQUIRY_OUTPUT.                                 |

**PCM_OP_TCF_AAA_ACCOUNTING_ON Block**
This block contains the fields shown in Table 3–14.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| INPUT   | Block that contains the input flist fields for this opcode.  
          | Data type is PCM_OP_TCF_AAA_ACCOUNTING_ON_INPUT.                                  |
| OUTPUT  | Block that contains the output flist fields for this opcode.  
          | Data type is PCM_OP_TCF_AAA_ACCOUNTING_ON_OUTPUT.                                 |

For information about fields and blocks in the INPUT and OUTPUT blocks, see the Pipeline_home/formatDesc/Formats/Portal/aaaContainerDesc.dsc file.
PCM_OP_TCF_AAA_ACCOUNTING_OFF Block
This block contains the fields shown in Table 3–15.

Table 3–15  PCM_OP_TCF_AAA_ACCOUNTING_OFF Block Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>Block that contains the input flist fields for this opcode.</td>
</tr>
<tr>
<td></td>
<td>Data type is PCM_OP_TCF_AAA_ACCOUNTING_OFF_INPUT.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Block that contains the output flist fields for this opcode.</td>
</tr>
<tr>
<td></td>
<td>Data type is PCM_OP_TCF_AAA_ACCOUNTING_OFF_OUTPUT.</td>
</tr>
</tbody>
</table>

For information about fields and blocks in the INPUT and OUTPUT blocks, see the Pipeline_home/formatDesc/Formats/Portal/aaaContainerDesc.dsc file.

PCM_OP_TCF_AAA_REFUND Block
This block contains the fields shown in Table 3–16.

Table 3–16  PCM_OP_TCF_AAA_REFUND Block Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>Block that contains the input flist fields for this opcode.</td>
</tr>
<tr>
<td></td>
<td>Data type is PCM_OP_TCF_AAA_REFUND_INPUT.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Block that contains the output flist fields for this opcode.</td>
</tr>
<tr>
<td></td>
<td>Data type is PCM_OP_TCF_AAA_REFUND_OUTPUT.</td>
</tr>
</tbody>
</table>

For information about fields and blocks in the INPUT and OUTPUT blocks, see the Pipeline_home/formatDesc/Formats/Portal/aaaContainerDesc.dsc file.

ASSOCIATED_PROTOCOL_INFO Block
This block contains the fields shown in Table 3–17.

Table 3–17  ASSOCIATED_PROTOCOL_INFO Block Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASS_MBI_INFO</td>
<td>Block containing all the fields in an MBI-based request, including fields</td>
</tr>
<tr>
<td></td>
<td>that have no equivalent in the input flists of the opcodes in the AAA</td>
</tr>
<tr>
<td></td>
<td>EDR description.</td>
</tr>
<tr>
<td></td>
<td>Data type is ASSOCIATED_MBI_INFO.</td>
</tr>
<tr>
<td>ASS_FLIST_INFO</td>
<td>Block containing all the fields in an flist-based request.</td>
</tr>
<tr>
<td></td>
<td>Data type is ASSOCIATED_FLIST_INFO.</td>
</tr>
<tr>
<td>ASS_OCSAC_INFO</td>
<td>Block containing all the fields in an MBI CDR required by the opcodes</td>
</tr>
<tr>
<td></td>
<td>that process the request.</td>
</tr>
<tr>
<td></td>
<td>Data type is ASSOCIATED_OCSAC_INFO.</td>
</tr>
<tr>
<td>ASS_DIAMETER_INFO</td>
<td>Block containing all the attribute/value pairs (AVPs) in a</td>
</tr>
<tr>
<td></td>
<td>Diameter-based request, including AVPs that have no equivalent field</td>
</tr>
<tr>
<td></td>
<td>in the input flists of the opcodes in the AAA EDR description.</td>
</tr>
<tr>
<td></td>
<td>Data type is ASSOCIATED_DIAMETER_INFO.</td>
</tr>
</tbody>
</table>
For information about fields in the Diameter and MBI protocol blocks, see the `Pipeline_home/formatDesc/Formats/Portal/aaaContainerDesc.dsc` file.

**PCM_OP_EBUF Block**

This block stores the contents of the `ebuf` error buffer after the opcode called to implement a request is executed. It contains the fields shown in Table 3–18. For more information about these fields, including possible values, see *BRM Developer’s Guide*.

<table>
<thead>
<tr>
<th>Field</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION</td>
<td>Integer</td>
<td>BRM module that encountered the error.</td>
</tr>
<tr>
<td>PIN_ERRCLASS</td>
<td>Integer</td>
<td>Class of error that occurred.</td>
</tr>
<tr>
<td>PIN_ERROR</td>
<td>Integer</td>
<td>Error that occurred.</td>
</tr>
<tr>
<td>FIELD</td>
<td>Integer</td>
<td>Field number of input parameter that caused the error.</td>
</tr>
<tr>
<td>REC_ID</td>
<td>Integer</td>
<td>Element ID of the array element that caused the error.</td>
</tr>
<tr>
<td>RESERVED</td>
<td>Integer</td>
<td>Internal system state used by Oracle for debug purposes.</td>
</tr>
<tr>
<td>FACILITY</td>
<td>Integer</td>
<td>Specifies the code of a facility associated with BRM internationalization (I18N) features. Used with the MSG_ID value to create a localized error message.</td>
</tr>
<tr>
<td>MSG_ID</td>
<td>Integer</td>
<td>Specifies a unique ID number for each message within the facility identified by the FACILITY code. Used with the FACILITY value to create a localized error message.</td>
</tr>
<tr>
<td>ERR_TIME_SEC</td>
<td>Integer</td>
<td>Time in seconds when the error occurred.</td>
</tr>
<tr>
<td>ERR_TIME_USEC</td>
<td>Integer</td>
<td>Time in microseconds when the error occurred.</td>
</tr>
<tr>
<td>PIN_VERSION</td>
<td>Integer</td>
<td>Version of the arguments.</td>
</tr>
<tr>
<td>FIELDLIST</td>
<td>String</td>
<td>Pipeline module that generated the error.</td>
</tr>
<tr>
<td>RESERVED2</td>
<td>Integer</td>
<td>Error type.</td>
</tr>
</tbody>
</table>

**INTERNAL Record in AAA EDRs**

The INTERNAL record in the AAA EDR container description defines a block of fields that appears in all AAA EDR SERVICE container types. The information in these fields is used by the pipeline framework to manage system operations for AAA Gateway Manager, such as starting and stopping pipeline transactions.

This record contains the fields shown in Table 3–19.
Table 3–19  INTERNAL Record Fields in AAA EDRs

<table>
<thead>
<tr>
<th>Field</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STREAM_NAME</td>
<td>String</td>
<td>For real-time pipelines, specifies the name of the output stream that the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>message being processed is sent to.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For batch pipelines, specifies the name of the file being processed.</td>
</tr>
<tr>
<td>OFFSET_GENERATION</td>
<td>Integer</td>
<td>For batch pipelines, used by the output controller when it performs sequence</td>
</tr>
<tr>
<td></td>
<td></td>
<td>checking for recycled EDRs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used by pipelines that read from a socket.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See “About Sequence Checking” in BRM System Administrator’s Guide for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>more information.</td>
</tr>
<tr>
<td>SEQ_CHECK</td>
<td>Integer</td>
<td>For batch pipelines, specifies whether sequence checking is enabled for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the current file. If yes, set to True. If no, set to False.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used by pipelines that read from a socket.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See “About Sequence Checking” in BRM System Administrator’s Guide for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>more information.</td>
</tr>
<tr>
<td>SEQ_GENERATION</td>
<td>Integer</td>
<td>For batch pipelines, specifies whether sequence generation is enabled for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the current file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ True = Sequence generation is enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ False = Sequence generation is not enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used by pipelines that read from a socket.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See “About Sequence Generation” in BRM System Administrator’s Guide for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>more information.</td>
</tr>
<tr>
<td>TRANSACTION_ID</td>
<td>Decimal</td>
<td>For batch pipelines, specifies the transaction ID of the set of input files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to which the current file belongs. This ID can be used to commit or roll back</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the entire set of files.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used by pipelines that read from a socket.</td>
</tr>
<tr>
<td>PROCESS_STATUS</td>
<td>Integer</td>
<td>For batch pipelines, specifies whether the current file is being recycled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(If the file was previously processed, sequence checking doesn’t need to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>be performed.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Can be set to one of these values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 0 (default) = normal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 1 = recycling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 2 = recycling test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used by pipelines that read from a socket.</td>
</tr>
<tr>
<td>PIN_FLD_POID</td>
<td>POID</td>
<td>Specifies the routing POID for a client’s logon request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used internally to make Connection Manager (CM) connections. Not set in EDRs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>that are passed through a pipeline.</td>
</tr>
<tr>
<td>PIN_FLD_RESULT</td>
<td>Integer</td>
<td>Specifies whether a client’s logon attempt is successful.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used internally to make CM connections. Not set in EDRs that are passed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>through a pipeline.</td>
</tr>
<tr>
<td>PIN_FLD_TYPE</td>
<td>Integer</td>
<td>Specifies whether a password is required for a client to log on.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used internally to make CM connections. Not set in EDRs that are passed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>through a pipeline.</td>
</tr>
<tr>
<td>PIN_FLD_PROGRAM_NAME</td>
<td>String</td>
<td>Specifies the name of the client that is logging on.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used internally to make CM connections. Not set in EDRs that are passed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>through a pipeline.</td>
</tr>
<tr>
<td>PIN_FLD_LOGIN</td>
<td>String</td>
<td>Specifies a client’s logon user ID.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used internally to make CM connections. Not set in EDRs that are passed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>through a pipeline.</td>
</tr>
</tbody>
</table>
When AAA Gateway Manager processes network requests, the opcodes it calls, the information it passes to the opcodes, and the information it receives from the opcodes and sends back to the network are determined by the AAA EDR container description. By default, this description supports fields in MBI-based requests that are required to perform AAA for GSM services. See "AAA Gateway Manager Protocol Support" for more information.

To pass more information to and from the default opcodes, to call additional opcodes, or to support other services, you must customize the description and its associated grammar and mapping files.

To modify the AAA EDR container description, see "Modifying and Loading the EDR Container Description" in BRM Setting Up Pricing and Rating for more information.

To modify the associated grammar and mapping files, see these topics:

- Configuring EDR Input Processing in BRM Configuring Pipeline Rating and Discounting
- Configuring EDR Output Processing in BRM Configuring Pipeline Rating and Discounting

<table>
<thead>
<tr>
<th>Field</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_PSSWD_CLEAR</td>
<td>String</td>
<td>Specifies a client’s logon password. Used internally to make CM connections. Not set in EDRs that are passed through a pipeline.</td>
</tr>
<tr>
<td>PIN_FLD_ACTION</td>
<td>String</td>
<td>Specifies the operation that a client is logging on to perform. Used internally to make CM connections. Not set in EDRs that are passed through a pipeline.</td>
</tr>
<tr>
<td>PIN_FLD_SESSION_OBJ</td>
<td>POID</td>
<td>Specifies the ID of the session established when a client logs on. Used internally to make CM connections. Not set in EDRs that are passed through a pipeline.</td>
</tr>
<tr>
<td>PIN_FLD_END_T</td>
<td>Date</td>
<td>Specifies the end time of the session established when a client logged on. Used internally to make CM connections. Not set in EDRs that are passed through a pipeline.</td>
</tr>
<tr>
<td>REACTOR_IDPRE</td>
<td>Integer</td>
<td>First part of the reactor’s address.</td>
</tr>
<tr>
<td>REACTOR_IDPOST</td>
<td>Integer</td>
<td>Second part of the reactor’s address.</td>
</tr>
</tbody>
</table>