Oracle® Communications Billing and Revenue Management

Synchronization Queue Manager

Release 12.0
E51032-03
April 2024

ORACLE

Oracle Communications Billing and Revenue Management Synchronization Queue Manager, Release 12.0

E51032-03

Contents

Preface

This guide describes how to use the Oracle Communications Billing and Revenue Management (BRM) Synchronization Queue data manager (DM) to synchronize pricing data between BRM and external customer relationship management applications.

Audience

This guide is intended for developers and system administrators.

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 that have purchased support 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.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

Understanding the Synchronization Queue Data Manager

This document describes the Oracle Communications Billing and Revenue Management (BRM) Synchronization Queue Data Manager (DM) and how it works.

See also:

- Installing and Configuring the Synchronization Queue DM
- Exporting Pricing Data in a Batch

About Synchronizing Pricing Data

You can synchronize the following pricing data between BRM and external customer relationship management (CRM) applications:

- Charge offers, including their associated provisioning tag details.
- Simple discounts in discount offers. See "About Simple Discounts".
- Real-time chargeshare offers.

When you start a CRM integration, you can export all of your existing charge offers, discount offers, and chargeshare offers from BRM to the external CRM application in a batch by using the **pin_export_price** utility. After BRM and the external CRM application are integrated, BRM can send charge offer, discount offer, and chargeshare offer changes to the external CRM application in real time.

> **Note:**
>
> For information about sending account data updates from BRM to Oracle Communications Elastic Charging Engine (ECE), see "About Synchronizing Account Data between the BRM Database and ECE" in *BRM Installation Guide*.

About Simple Discounts

BRM evaluates discounts in discount offers when they are created or modified to determine whether they are simple or complex. BRM considers a discount to be simple if it meets the following criteria:

- The discount mode is original charge or remaining charge.
- The event type is a purchase or cycle event.
- The discount trigger condition is always true: (1 >= 0) and the condition expression is an integer value.
- The quantity range expression is “Charge".

- The discount contains only one discount rule, one discount quantity range, and one discount balance impact.
- The discount balance impact is set to the impact currency balance element.
- The base expression is “Charge".

If a discount is considered simple, BRM sends the following information to the external CRM application: the balance element ID, the discount amount, and a flag indicating whether the discount amount is a percentage (**P**) or an absolute (**A**) value.

If a discount is not considered simple, BRM sends to the external CRM application only the balance element ID and the discount amount, which is set to 0.

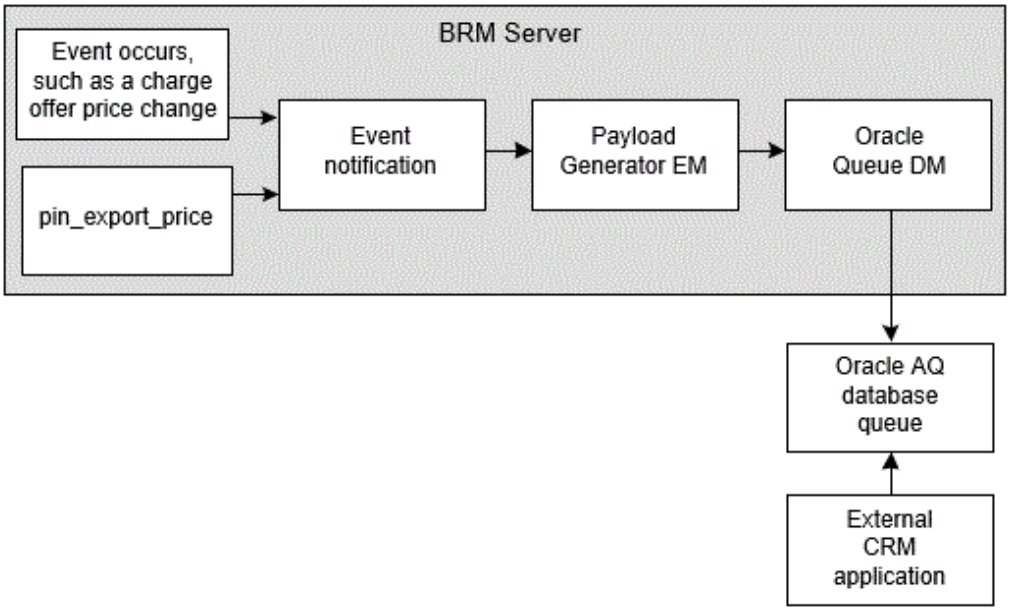
About the Data Synchronization Process

BRM synchronizes charge offers, discount offers, and chargeshare offers by using the Synchronization Queue DM and the Enterprise Applications Integration (EAI) framework, which consists of the event notification system and the Payload Generator External Module (EM). The Synchronization Queue DM and EAI framework work together to publish changes to a central Oracle Advanced Queuing (AQ) database queue called the Synchronization Queue DM database queue. External CRM applications can then retrieve data directly from the database queue.

> **Note:**
>
> To use the Synchronization Queue DM, you must purchase and install EAI Manager. For more information, see "About Integrating BRM with Enterprise Applications" in *BRM Developer's Guide*.

Figure 1-1 shows the data flow from BRM to the external CRM application:

Figure 1-1 Data Synchronization Process



Charge offers, discount offers, and chargeshare offers are synchronized in the following way:

1. A notification event is generated. The way a notification event is generated depends on whether you are synchronizing in real time or in a batch:
   - **In real time**: BRM generates notification events right after any of the following occurs: a charge offer changes, a discount offer changes, or a chargeshare offer changes.
   - **In a batch**: The **pin_export_price** utility generates the notification events. When you run the utility, it retrieves charge offers, discount offers, and chargeshare offers from the BRM database and then generates a notification event for each object that is retrieved. See "Exporting Pricing Data in a Batch".
2. The BRM event notification system recognizes the notification event and calls the Payload Generator EM. See "About BRM Event Notification".
3. The Payload Generator EM collects events in its payload until they compose a complete business event. See "About the Payload Generator EM".
4. When the business event is complete, the Payload Generator EM sends it to the Synchronization Queue DM. See "About the Synchronization Queue DM".
5. The Synchronization Queue DM sends the business event to the Synchronization Queue DM database queue. See "About the Synchronization Queue DM Database Queue".
6. The external CRM application retrieves the business event from the database queue and updates the information in its system.

About BRM Event Notification

The following notification events are generated when you run the **pin_export_price** utility or when charge offers, discount offers, and chargeshare offers change in the BRM system:

- **/event/notification/price/discounts/modify**
- **/event/notification/price/products/modify**
- **/event/notification/price/sponsorships/modify**

When BRM is configured to synchronize charge offers, discount offers, and chargeshare offers, the event notification system listens for one of these events to occur and then calls the Payload Generator EM.

You define the notification events that trigger calls to the Payload Generator EM by using the *BRM_home***/sys/data/config/pin_notify_crm_sync** file. See "Configuring Event Notification for the Synchronization Queue DM".

For more information about event notification, see "About Event Notification" in *BRM Developer's Guide*.

About the Payload Generator EM

The Payload Generator EM collects notification events, generates the data necessary to publish business events, and sends the data to the Synchronization Queue DM.

The data required to create a complete business event is defined in the Synchronization Queue DM payload configuration file (*BRM_home***/sys/eai_js/payloadconfig_crm_sync.xml**). The default file includes definitions for the following business events:

- **ProductInfoChange** - Defines the events related to charge offer information changes, such as charge offer pricing changes.
- **DiscountInfoChange** - Defines the events related to discount offer information changes, such as discount offer criteria changes.
- **SponsorshipInfoChange** - Defines the events related to chargeshare offer (***/sponsorship*** object) changes.

You can add custom business events to the payload configuration file. For information on creating business events, see "Defining Business Events" in *BRM Developer's Guide*.

For information on how to configure the **payloadconfig_crm_sync** file, see "Configuring the EAI Payload for the Synchronization Queue DM".

About the Synchronization Queue DM

The Synchronization Queue DM is responsible for publishing data to the Synchronization Queue DM database queue. You define which business events are sent to the database queue by using the queue-names file (*BRM_home***/sys/dm_aq/aq_queuenames**). To configure the queue-names file, see "Specifying Which Business Events to Send to the Database Queue".

When the Synchronization Queue DM receives a business event from the Payload Generator EM, it does the following:

1. Determines whether the event should be sent to the Synchronization Queue DM database queue by checking the **aq_queuenames** file.
2. Publishes the entire contents of a business event to the database queue as an XML message.
3. Sets the event in the database queue to a READY state. For information, see "About Event Status Flags".

> **Note:**
>
> If the Synchronization Queue DM is offline for an extended period of time, you can use the **pin_export_price** utility to synchronize charge offer and discount offer CRM data in batches. For more information, see "Exporting Pricing Data in a Batch".

About the Synchronization Queue DM Database Queue

The Synchronization Queue DM database queue is used to pass business events from the Synchronization Queue DM to the external CRM application. The Synchronization Queue DM sends business events to the database queue asynchronously, so BRM and the external CRM application are not required to be running at the same time.

- If the external CRM application terminates, the Synchronization Queue DM continues to send business events to the database queue. After the external CRM application restarts, it retrieves the business events from the database queue.
- If the Synchronization Queue DM terminates, the external CRM application continues to retrieve business events already in the database queue.

About the Structure of Queued Messages

The Synchronization Queue DM publishes the entire contents of a business event. The business event is stored as a message in a **/deq_event_ty** object. This object includes the following fields:

- PIN_FLD_EVENT_NAME — Specifies the name of the business event. The external CRM application can use this event name to identify the type of data in the queued message. For more information, see "About Retrieving Specific Events from the Synchronization Queue DM Database Queue".
- PIN_FLD_FLIST_BUF — Contains the message (business event) in XML format if the message size is 4000 bytes or less.
- PIN_FLD_LARGE_FLIST_BUF — Contains the message (business event) in XML format if the message size is greater than 4000 bytes.
- PIN_FLD_MESG_ID — Contains the unique message ID generated by the Synchronization Queue DM database queue.
- PIN_FLD_ENQ_TIME — Specifies the time the message was sent to the Synchronization Queue DM database queue.

About Event Status Flags

Events in the Synchronization Queue DM database queue are set to the following states:

- **READY** indicates that the event has not been retrieved by the external CRM application.
- **PROCESSED** indicates that the event was retrieved by the external CRM application. The Oracle Queue Monitor process (QMn) removes the event from the database queue after a configurable amount of time.

You can check the status of events in the Synchronization Queue DM database queue by running a report. For information, see "Generating Queue Reports".

About Retrieving Specific Events from the Synchronization Queue DM Database Queue

Business event names are defined in the base tags of the payload configuration file (**payloadconfig_crm_sync.xml**). When an event is sent to the Synchronization Queue DM database queue, the business event name is stored in the database queue table in the CORRID field. The external CRM application can check for the name in the CORRID field to identify the type of data in the queued messages and determine whether to retrieve the message.

For example, if the name of a business event is **ProductInfoChange**, the message contains information about changes to a charge offer. If the external CRM application needs to update its charge offer information, it can retrieve the event. Otherwise, it can ignore the event.

> **Note:**
>
> The CORRID field is limited to 128 characters. If you customize the **payloadconfig_crm_sync.xml** file, be sure to use business event names that have fewer than 128 characters.

2

Installing and Configuring the Synchronization Queue DM

This document describes how to install and configure Oracle Communications Billing and Revenue Management (BRM) Synchronization Queue Data Manager (DM).

See also:

- Understanding the Synchronization Queue Data Manager
- Exporting Pricing Data in a Batch

> **Note:**
>
> Synchronization Queue Manager synchronizes pricing data in BRM and external customer relationship management (CRM) applications. For information about installing and configuring account synchronization, which synchronizes account data in BRM and Elastic Charging Engine (ECE), see "Installing and Configuring Account Synchronization" in *BRM Installation Guide*.

About Installing the Synchronization Queue DM

Before setting up the Synchronization Queue DM, you should be familiar with BRM concepts and architecture. See "About BRM" and "BRM System Architecture" in *BRM Concepts*.

> **Note:**
>
> Synchronization Queue DM is included in two separate downloadable packages: Web Services Manager and EAI Manager.

Installing and Configuring the Synchronization Queue DM

> **Note:**
>
> Install the Synchronization Queue DM and then perform *all* configuration steps before starting the Synchronization Queue DM.

Installing the Synchronization Queue DM requires you to:

1. Configure the database for Oracle Advanced Queuing (AQ). See "Configuring the Database for Advanced Queuing".
2. Install the Synchronization Queue DM. See "Installing the Synchronization Queue DM".
3. If your database queue is not located on the system where BRM is installed, link the databases. See "Linking Oracle Databases".
4. Configure the EAI payload for synchronization queuing. See "Configuring the EAI Payload for the Synchronization Queue DM".
5. Enable event notification. See "Configuring Event Notification for the Synchronization Queue DM".
6. Configure the Synchronization Queue DM. See "Configuring the Synchronization Queue DM".

Configuring the Database for Advanced Queuing

Before you install the Synchronization Queue DM, you must configure the queuing database machine for Oracle Advanced Queuing (AQ).

Perform the following on the queuing database machine in your system:

1. Open the *Oracle_home*/**dbs/init.ora** file.
2. Set the **compatible** parameter to the Oracle database version.

> **Note:**
>
> If you use an entry lower than **10.0**, your BRM system fails.

3. Set the **aq_tm_process** parameter to **1** to specify one Oracle Queue Monitor process (QMn). QMn removes from the queue any processed events that are over an hour old.
4. Save and close the file.
5. Using SQL*Plus, log in to your database as the SYS user and grant AQ privileges to user **pin**:

```
% sqlplus sys@databaseAlias as sysdba
Enter password: password

SQL> grant execute on dbms_aq to pin;

Grant succeeded.

SQL> grant execute on dbms_aqadm to pin;

Grant succeeded.

SQL> grant select on sys.gv_$aq to pin;

Grant succeeded.

SQL> grant execute on dbms_lock to pin;

Grant succeeded.
```

```
SQL> grant execute on dbms_aqin to pin;

Grant succeeded.

SQL> grant execute on sys.dbms_aqin to pin;

Grant succeeded.
```

6. Stop and restart the Oracle database, which initializes the database instance with your changes.

Installing the Synchronization Queue DM

To install Synchronization Queue DM, see "Installing Individual BRM Components" in *BRM Installation Guide*.

> **Note:**
>
> If you already installed the product, you must uninstall its features before reinstalling them.

Linking Oracle Databases

> **Note:**
>
> If your database queue resides on the system on which BRM is installed, skip this section and go to "Configuring the EAI Payload for the Synchronization Queue DM".

If the database queue is installed on a system other than the system on which BRM is installed, you must create a link from the queuing database to the BRM database. This enables Oracle to forward events to the queue residing on the other database.

Before linking your Oracle databases, verify that the database alias names are defined in each **tnsnames.ora** file (*Oracle_home*/**network/admin/tnsnames.ora**).

> **Tip:**
>
> You can check this by logging in to one database as user **pin** and connecting to the other databases by using the database alias. For example:
>
> ```
> SQL> connect pin@databaseAlias
> Enter password: password
> ```

> **Note:**
>
> To change the location of the database queue, follow the instructions in "Manually Creating a Database Queue on an Oracle Database".

To link your databases, perform the following steps from each database:

1. Using SQL*Plus, log in to your database as the SYSTEM user and grant database linking privileges to user **pin**:

```
% sqlplus system@databaseAlias
Enter password: password

SQL> grant create database link to pin;

Grant succeeded.
```

2. Log in to the database as user **pin**:

```
SQL> connect pin@databaseAlias
Enter password: password
```

3. Create a link to the other databases by entering the following command:

```
SQL> CREATE DATABASE LINK remoteDatabaseAlias CONNECT TO pin IDENTIFIED BY
pin_password USING 'RemoteDatabaseAlias';
```

   For example, if your BRM database has the alias name **Portal1** and the database where the queue is located has the alias name **Portal2**, log in to **Portal1** and give the following SQL command:

```
SQL> CREATE DATABASE LINK portal2 CONNECT TO pin IDENTIFIED BY pin_password
USING 'portal2';
```

   and log in to **Portal2** and give the following SQL command:

```
SQL> CREATE DATABASE LINK portal1 CONNECT TO pin IDENTIFIED BY pin_password
USING 'portal1';
```

Configuring the EAI Payload for the Synchronization Queue DM

You must modify your BRM system's payload configuration file in the following situations:

- If you have another EAI-based publisher, you must do the following.

> **Tip:**
>
> The name and location of any existing file are specified in the **infranet.eai.configFile** entry in the EAI properties file (*BRM_home***/sys/eai_js/Infranet.properties**).

  - Check for conflicts in the EAI payload configuration files. See "Checking for Conflicts in EAI payload Configuration Files".
  - Specify the default configuration file. See "Specifying the Default Payload Configuration File".
- If your Synchronization Queue DM database number is not the default (0.0.9.7), see "Specifying the Synchronization Queue DM Database Number".
- If you uninstall the Synchronization Queue DM, see "Revising the Payload Configuration File When Uninstalling the Synchronization Queue DM".

For information about defining business events in the EAI payload configuration file, see "About the Payload Configuration File Syntax" in *BRM Developer's Guide*.

Checking for Conflicts in EAI payload Configuration Files

If your BRM system already has an EAI publisher, the Synchronization Queue DM installation program merges the Synchronization Queue DM payload configuration file with the existing payload configuration file that is referenced in *BRM_home***/sys/eai_js/Infranet.properties**. The merged file is named **payloadconfig_MergedWithCRMSync.xml**.

In rare cases, conflicts can occur when payload configuration files are merged. This can happen when entries in the two files have conflicting definitions.

> **Note:**
>
> You must determine if there are merge conflicts *before* you run the Synchronization Queue DM.

Compare the Synchronization Queue DM payload configuration file (*BRM_home***/sys/eai_js/payloadconfig_crm_sync.xml**) with the existing payload configuration file. If any of the following conditions exists, you must modify the business events in conflict and manually merge the two files in order to run the Synchronization Queue DM with other EAI publishers in the same BRM system:

- Two different business event definitions specify the same **StartEvent**.
- The same business event definitions have different values for one or more of these attributes: **StartEvent**, **EndEvent**, or **Source**.
- The same business event or element definitions have different search criteria.
- The same element definitions have different values for one or more of these attributes: **Source**, **PinFld**, **DataFrom**, **UseOnlyElement**, or **Tag**.
- The same element definitions have different **OnEvent** values.

Specifying the Default Payload Configuration File

If your BRM system already has an EAI publisher, you must make the merged payload configuration file your system's default payload configuration file.

To specify the default payload configuration file, change the EAI **configFile** entry so that it points to the merged payload configuration file:

1. In a text editor, open the **eai_js** properties file (*BRM_home*/**sys/eai_js/Infranet.properties**).
2. Change the **infranet.eai.configFile** entry to refer to the merged file:

   ```
   infranet.eai.configFile=./payloadconfig_file_name.xml
   ```

   where **./payloadconfig_***file_name***.xml** is the name of the merged file.

   The default merged file name is **payloadconfig_MergedWithCRMSync.xml**. If you manually merged payload configuration files, the file name is the name you gave it.

   > **Tip:**
   >
   > You can use any meaningful name for the file that you choose, providing that it starts with **payloadconfig_** and that the file name and the value of the **infranet.eai.configFile** entry match.

3. Save and close the file.

Specifying the Synchronization Queue DM Database Number

The default BRM database number for your Synchronization Queue DM is 0.0.9.7. If you use a different database number, you must change the value of the publisher database in the payload configuration file.

1. In a text editor, open the payload configuration file in the *BRM_home*/**sys/eai_js** directory (**payloadconfig_crm_sync.xml**, or the merged file if you merged payload configuration files).
2. Find the **<PublisherDefs>** section.
3. In the **Publisher DB** entry, specify the correct BRM database number.

   For example, if your Synchronization Queue DM database number is 0.0.9.5, change this entry:

   ```
   <PublisherDefs>
       <Publisher DB="0.0.9.7" Format="XML">
   ```

   to this:

   ```
   <PublisherDefs>
       <Publisher DB="0.0.9.5" Format="XML">
   ```

4. Save and close the file.

Revising the Payload Configuration File When Uninstalling the Synchronization Queue DM

To remove Synchronization Queue elements from your system's EAI payload configuration file after uninstalling the Synchronization Queue DM:

1. In a text editor, open the payload configuration file in the *BRM_home*/**sys/eai_js** directory (**payloadconfig_crm_sync.xml**, or the merged file if you merged payload configuration files).
2. Find the **<PublisherDefs>** section.
3. Remove the following publisher definition:

```
<Publisher DB="0.0.9.7" Format="XML">
```

> **Note:**
>
> This is the default Synchronization Queue DM publisher definition. Your definition might have a different **Publisher DB** value.

4. Save and close the file.
5. Stop and restart the Payload Generator External Module (the EAI Java server) by entering this command from the *BRM_home*/**bin** directory:

```
pin_ctl restart eai_js
```

Configuring Event Notification for the Synchronization Queue DM

Business events defined in the Synchronization Queue DM payload configuration file include a set of BRM events. When one of these BRM events occurs, the EAI framework uses event notification to call the opcode that caches the BRM event in the Payload Generator.

Before you can use the Synchronization Queue DM, you must configure the event notification feature as follows:

1. If your system has multiple configuration files for event notification, merge them.
2. Ensure that the merged file includes the entire event notification list in the *BRM_home*/**sys/data/config/pin_notify_crm_sync** file.
3. (Optional) If you defined new business events in the Synchronization Queue DM payload configuration file, you must edit your final event notification list to include all the BRM events in the new business events.
4. Load your final event notification list into the BRM database.

For more information, see "Using Event Notification" in *BRM Developer's Guide*.

Configuring the Synchronization Queue DM

You configure the Synchronization Queue DM to connect to the BRM database and the database queue by performing these tasks:

1. Connect the Synchronization Queue DM's EAI framework to BRM by editing the CM configuration file (**pin.conf**). See "Configuring the CM for Synchronization Queuing".
2. Map BRM business events to the database queue by editing the **aq_queuenames** file. See "Specifying Which Business Events to Send to the Database Queue".
3. Connect the Synchronization Queue DM to the database queue by editing the Synchronization Queue DM configuration file (**pin.conf**). See "Configuring the Synchronization Queue DM Configuration File ".

Configuring the CM for Synchronization Queuing

You must modify the CM **pin.conf** file to enable the EAI framework to notify the Synchronization Queue DM when specific events occur. You should also verify that the pointer to the Synchronization Queue DM specifies the correct database and port numbers.

To configure the CM for synchronization queuing:

1. Open the CM configuration file (*BRM_home***/sys/cm/pin.conf**) in a text editor.
2. Set the **enable_publish** entry to **1**:

   ```
   - fm_publish enable_publish 1
   ```

3. Verify that the **dm_pointer** entry for the Synchronization Queue DM specifies the correct database and port:

   ```
   - cm dm_pointer db_number ip host_name_or_ip_address port
   ```

   where:

   - *db_number* is the Synchronization Queue DM database number. The default is **0.0.9.7**.

     > **Note:**
     >
     > The Synchronization Queue DM *database number* must match the number in these entries:
     >
     > – The **dm_db_no** entry in the Synchronization Queue DM configuration file (*BRM_home***/sys/dm_aq/pin.conf**).
     >
     > – The **Publisher DB** entry of the Synchronization Queue publisher definition in the payload configuration file (*BRM_home***/sys/eai_js/payloadconfig_crm_sync.xml**).

   - *host_name_or_ip_address* is the host name or IP address of the computer on which the Synchronization Queue DM runs.
   - *port* is the Synchronization Queue DM port number. The default is **17513**.

     > **Note:**
     >
     > The Synchronization Queue DM *port number* must match the number in the **dm_port** entry in the Synchronization Queue DM configuration file (*BRM_home***/sys/dm_aq/pin.conf**).

> **Note:**
>
> If you change the location of the Synchronization Queue DM, you need to modify this entry.

4. Save and close the file.
5. Stop and restart the CM.

Specifying Which Business Events to Send to the Database Queue

You must configure which events the Synchronization Queue DM sends to the database queue by editing the **aq_queuenames** file.

> **Note:**
>
> - Only business events that are defined in the **payloadconfig** file can be sent to the database queue.
> - The Synchronization Queue DM does not send events to the queue until you edit the **aq_queuenames** file.

To specify which business events to send to the database queue:

1. Open the *BRM_home*/**sys/dm_aq/aq_queuenames** file in a text editor.
2. Specify the business events to send to the queue:
   - If the queue resides on the database from which the Synchronization Queue DM connects, use this syntax:

     ```
     queue_name
     {
       business_events
     }
     ```

   - If the queue resides on a separate database from which the Synchronization Queue DM connects, use this syntax:

     ```
     queue_name@database_link
     {
       business_events
     }
     ```

   where:

   - *queue_name* specifies the queue name. The queue name must match the name you assigned to the queue when you created it.
   - *database_link* specifies the Oracle database link for connecting to the queue if it resides on another database. Use the database link names you created in "Linking Oracle Databases".
   - *business_events* specifies which events to send to the queue. You can configure the Synchronization Queue DM to send all business events defined in the

payloadconfig file or only specific business events. You can also specify to exclude events from being sent to the database queue. Table 2-1 shows the syntax for each criterion:

Table 2-1 Synchronization Queue Event Syntax

To send...	... use this syntax
All business events	**ALL**
Only specific business events	*event_name*
All business events except those specified	**!***event_name*

For example, to send all business events to the AQ_QUEUE queue:

```
AQ_QUEUE
{
  ALL
}
```

To send only **ProductInfoChange** and **DiscountInfoChange** events to the queue, separate the events by a space, a new line, or the word OR:

```
AQ_QUEUE
{
  ProductInfoChange
  DiscountInfoChange
}
```

To send all business events except **DiscountInfoChange** to the queue located on a database with the alias **Portal2**:

```
AQ_QUEUE@Portal2
{
  !DiscountInfoChange
}
```

3. Save and close the file.

Configuring the Synchronization Queue DM Configuration File

During installation, the Synchronization Queue DM installer generates a configuration file (**pin.conf**) that specifies how to connect to your BRM database and database queue and that contains other configuration settings. The installer populates the connection entries with values from your **pin_setup.values** file and provides default information for the other configuration entries.

Before you start the Synchronization Queue DM, verify that the configuration file contains accurate information. If necessary, change the values in the file to those specified in the following steps:

1. Open the Synchronization Queue DM configuration file (*BRM_home***/sys/dm_aq/pin.conf**) in a text editor.
2. Verify the **plugin_name** entry:

   -dm plugin_name *path*/*name*

where:

- *path* is the path to the shared library file.
- *name* is the shared library file name.

3. Verify the **dm_sm_obj** entry:

```
-dm dm_sm_obj path/name
```

where:

- *path* is the path of the shared library file for the EAI DM.
- *name* is the name of the shared library file, which is **libdm_eai.so**.

4. Verify the **queue_map_file** entry specifies the path and file that maps the Oracle database queue to the event it is to receive. For example:

```
- dm_ifw_sync queue_map_file ./aq_queuenames
```

For more information, see "Specifying Which Business Events to Send to the Database Queue".

5. Verify the **sm_database** entry:

```
- dm sm_database alias_name
```

where *alias_name* is the alias name of the Oracle queuing database to which the Synchronization Queue DM connects. The value of this entry should be the TNSNAMES.ORA alias, which can be found in the *Oracle_home***/network/admin/tnsnames.ora** file:

6. Verify the **sm_id** entry:

```
- dm sm_id user_name
```

where *user_name* specifies the database user name that the Synchronization Queue DM uses to log in to the queuing database.

7. Verify the **dm_db_no** entry:

```
- dm dm_db_no database
```

Where *database* is the number of the database on which **dm_aq** is installed. The default is **0.0.9.7 / 0**.

8. Verify the **dm_port** entry:

```
- dm dm_port port_number
```

where *port_number* is the database port for connecting to the DM. The default is **17513**.

9. (Optional) You can also edit these entries to change how often the Synchronization Queue DM attempts to connect to the database queue:

- **-dm_ifw_sync connect_retries**
- **-dm_ifw_sync retry_interval**

For more information, see "Configuring the Synchronization Queue DM Database Connection Attempts".

For other entries you can edit, see the comments in the **pin.conf** file.

10. Save and close the file.

Excluding Events Triggered by External Applications

Business events need to be sent to the Synchronization Queue DM database queue only when the transactions that generate business events originate with BRM. If a business event is generated by a transaction that originates with an external system, queuing the business event is unnecessary because the external system already knows about the change.

You can suppress business events that are generated due to actions taken on external systems from being published by specifying the external system's login service POID in the Synchronization Queue DM configuration file (**pin.conf**). The Synchronization Queue DM will not queue business events that include this login service POID.

To suppress business events for transactions that originate on external systems from being published:

1. Open the Synchronization Queue DM configuration file (*BRM_home*/**sys/dm_aq/pin.conf**) in a text editor.
2. Uncomment the **service_poid** entry and, if necessary, change the value of the **service_poid** entry. The default service is **/service/admin_client**. For example:

   ```
   -dm_ifw_sync service_poid 0.0.0.1 /service/admin_client 2
   ```

3. Save and close the file.

Starting and Stopping the Synchronization Queue DM

To start the Synchronization Queue DM, enter this command at the prompt:

```
pin_ctl start dm_aq
```

To stop the Synchronization Queue DM, enter this command at the prompt:

```
pin_ctl stop dm_aq
```

Monitoring and Maintaining the Synchronization Queues

This section provides information and guidelines to help you manage your Synchronization Queue DM database queues.

The main administrative tasks for database queues are:

- Manually Creating a Database Queue on an Oracle Database
- Generating Queue Reports
- Dropping the Queue and Queue Tables
- Configuring the Queue Location
- Configuring How Often Processed Events are Removed from the Queue
- Configuring the Synchronization Queue DM Database Connection Attempts
- Disconnecting and Reconnecting the Synchronization Queue DM to the Queue

Manually Creating a Database Queue on an Oracle Database

The Synchronization Queue DM installer automatically creates a default queue in a specified database. If you need to change the location of your queue or to re-create the queue, you can create a new queue by running the **pin_portal_sync_oracle** utility.

> **Note:**
>
> To avoid system errors, disconnect or stop your external applications that will connect to the queue you are adding before you run **pin_portal_sync_oracle**. After creating and testing your queue, you can reconnect the applications that will communicate with the queue.

When you create a queue, you must decide:

- The database in which to create the queue.

  You can create the queue in the BRM database or in an external database. The best location for the queue depends on your business needs: enqueuing performance will be better when the queue is on the BRM system; dequeuing performance will be better when the queue is on the external application's system.

- The tablespace in which to create the queue.

  You can create the queue in an existing tablespace or in its own separate tablespace.

  > **Tip:**
  >
  > For optimal performance in production systems, create the queue in its own tablespace.

To manually create a database queue:

1. Stop the Synchronization Queue DM if it is running. See "Starting and Stopping the Synchronization Queue DM".
2. If you are changing the location of the database queue, drop the existing queue. See "Dropping the Queue and Queue Tables".
3. If necessary, configure the Oracle database for synchronization queuing. See "Configuring the Database for Advanced Queuing".
4. If necessary, link the database where the queue will reside to the database where BRM is installed. See "Linking Oracle Databases".
5. Verify default queue storage and retention time settings in the **create_sync_queue.conf** file. For information, see:
   - Configuring the Queue Location
   - Configuring How Often Processed Events are Removed from the Queue
6. Create your queue by logging in as user **pin** and running **pin_portal_sync_oracle** with the **create** command:

> **Note:**
>
> To create queues, user **pin** must have Oracle AQ privileges.

```
% su - pin
% pin_portal_sync_oracle.pl create [-l username/password@databaseAlias] [-
q queue_name -t queue_table]
```

where:

-l specifies how to connect to the database.

-q specifies the name of the queue. If this parameter is not included, the utility creates a queue named AQ_QUEUE.

-t specifies the name of the queue database table. If this parameter is not included, the utility creates a table named AQ_SYNC.

See "pin_portal_sync_oracle" for more information about this utility.

7. Verify that the queue and queue table were created and function properly by running **pin_portal_sync_oracle** with the **test** command:

```
% pin_portal_sync_oracle.pl test [-q queue_name]
```

The utility queues and dequeues 20 test events.

8. Create a summary report to verify that the queue processed the test events properly by running **pin_portal_sync_oracle** with the **report** command:

```
% pin_portal_sync_oracle.pl report -r summary
```

If successful, the report shows all 20 events with a processed state:

```
Tue Feb 07                                              page 1
                 Event Summary Report
Evt. Stat  Event Name                    Event Count
---------  ----------------------------  -----------
PROCESSED  LongTestEvent                      10
           ShortTestEvent                     10
*********                                 -----------
sum                                            20
```

If unsuccessful, you must drop and then re-create the queue and queue table. For information, see "Dropping the Queue and Queue Tables".

9. If necessary, update the queuing database number. See "Specifying the Synchronization Queue DM Database Number".

You can now start the Synchronization Queue DM. See "Starting and Stopping the Synchronization Queue DM".

Generating Queue Reports

You can monitor the events in your database queue by running **pin_portal_sync_oracle** to generate reports.

> **Note:**
>
> To avoid system errors, disconnect or stop your external applications that will connect to the queue you are adding before you run **pin_portal_sync_oracle**. After creating and testing your queue, you can reconnect the applications that will communicate with the queue.

- A summary report lists the number of events set to the READY and PROCESSED states.
- A detailed report lists each event's ID, state, queuing time, and dequeuing time.

Generating Synchronization Queue DM reports

To create a summary report for Synchronization Queue DM, run **pin_portal_sync_oracle** using this command:

```
% pin_portal_sync_oracle.pl report -r summary [-q queue_name]
```

where *queue_name* specifies the queue name.

To run a detailed report for Synchronization Queue DM, run **pin_portal_sync_oracle** using this command:

```
% pin_portal_sync_oracle.pl report -r detail [-q queue_name]
```

where *queue_name* specifies the queue name.

For information, see "pin_portal_sync_oracle".

Dropping the Queue and Queue Tables

To drop the queue and its queue table, run **pin_portal_sync_oracle** with the **drop** command:

> **Note:**
>
> To avoid system errors, disconnect or stop your external applications that will connect to the queue you are adding before you run **pin_portal_sync_oracle**. After creating and testing your queue, you can reconnect the applications that will communicate with the queue.

```
% pin_portal_sync_oracle.pl drop [-q queue_name] [-l username/
password@DatabaseAlias]
```

where:

queue_name specifies the queue name.

-l specifies how to connect to the database.

The specified queue is dropped from the database and the utility removes the Synchronization Queue DM package, which contains stored procedures for queuing, dequeuing, and purging events.

For more information, see "pin_portal_sync_oracle".

Configuring the Queue Location

By default, **pin_portal_sync_oracle** creates a database queue in the tablespace specified when the Synchronization Queue DM was installed. To use a different tablespace, perform one of the following:

- Specifying Default Storage Settings by Using the create_sync_queue.conf File
- Specifying Storage Settings by Using the pin_portal_sync_oracle Utility

Specifying Default Storage Settings by Using the create_sync_queue.conf File

You can specify the default storage settings by using the **create_sync_queue.conf** file. The Synchronization Queue DM database queue that you create uses the default settings unless you override them with **pin_portal_sync_oracle**.

To specify your default storage settings:

1. Open the *BRM_home***/apps/pin_aq/create_sync_queue.conf** file in a text editor.
2. Specify the target tablespace and queue size by editing the **storage_clause** entry:

> **Tip:**
>
> For production systems, create your queue in its own, separate tablespace to improve processing performance.

```
$storage_clause = "tablespace PIN00 initrans 5 storage (initial 200k next
200k maxextents unlimited pctincrease 0 )";
```

3. Save and close the file.

Specifying Storage Settings by Using the pin_portal_sync_oracle Utility

You can specify the queue's storage settings by using **pin_portal_sync_oracle** with the **create** command and the **-s** parameter. This option overrides the storage settings in the **create_sync_queue.conf** file.

> **Note:**
>
> To avoid system errors, disconnect or stop your external applications that will connect to the queue you are adding before you run **pin_portal_sync_oracle**. After creating and testing your queue, you can reconnect the applications that will communicate with the queue.

To specify storage settings by using **pin_portal_sync_oracle**:

```
% su - pin
% pin_portal_sync_oracle.pl create [-q queue_name -t queue_table] -s storage_clause
```

where:

- *queue_name* specifies the queue name.
- *queue_table* specifies the queue table name.
- *storage_clause* specifies the queue's storage parameters.

For more information, see "pin_portal_sync_oracle".

> **Tip:**
>
> For production systems, create your queue in its own, separate tablespace to improve processing performance.

Configuring How Often Processed Events are Removed from the Queue

The Oracle Queue Monitor process (QMn) removes from the queue any event set to the PROCESSED state for a specified amount of time. A default retention time was specified when the Synchronization Queue DM was installed. To use a different retention time, perform one of the following tasks:

- Modify **pin_portal_sync_oracle** configuration file. See "Specifying Default Retention Times by Using the create_sync_queue.conf File".
- Specify the retention time when running **pin_portal_sync_oracle**. See "Specifying Retention Times by Using the pin_portal_sync_oracle Utility".

Specifying Default Retention Times by Using the create_sync_queue.conf File

You can specify the default retention time setting by using the **create_sync_queue.conf** file. The Synchronization Queue DM database queue that you create uses this default setting unless you override it with **pin_portal_sync_oracle**.

To specify how often processed events are removed:

1. Open the *BRM_home***/apps/pin_aq/create_sync_queue.conf** file in a text editor.
2. Set the **retention_time** parameter to the amount of time, in seconds, that you want to store processed events in the database queue:

> **Tip:**
>
> For production systems, set the retention time to **0** to optimize your processing performance.

```
$retention_time = retention_time;
```

3. Save and close the file.

Specifying Retention Times by Using the pin_portal_sync_oracle Utility

You can specify a queue's retention time by using **pin_portal_sync_oracle** with the **create** command and the **-r** parameter. This option overrides the retention time settings in the **create_sync_queue.conf** file.

> **Note:**
>
> To avoid system errors, disconnect or stop your external applications that will connect to the queue you are adding before you run **pin_portal_sync_oracle**. After creating and testing your queue, you can reconnect the applications that will communicate with the queue.

To set the retention time by using **pin_portal_sync_oracle**:

```
% su - pin
% pin_portal_sync_oracle.pl create [-q queue_name -t queue_table] -r
retention_time
```

where:

- *queue_name* specifies the queue name.
- *queue_table* specifies the queue table name.
- *retention_time* specifies the queue's retention time, in seconds.

For more information, see "pin_portal_sync_oracle".

> **Tip:**
>
> For production systems, set the retention time to **0** to optimize your processing performance.

Configuring the Synchronization Queue DM Database Connection Attempts

You can configure how often the Synchronization Queue DM attempts to connect to the database that contains the Synchronization Queue DM queue.

To configure connection attempts:

1. Open the Synchronization Queue DM configuration file (*BRM_home*/**sys/dm_aq/pin.conf**) in a text editor.
2. Specify the number of times the Synchronization Queue DM should retry to connect to the Oracle database server by editing the **connect_retries** entry. The default is **1**.

   ```
   -dm_aq connect_retries number_of_retries
   ```

3. Specify the interval, in seconds, between each reconnection attempt by editing the **retry_interval** entry. The default is **0**.

   ```
   -dm_aq retry_interval interval
   ```

4. Save and close the file.
5. Stop and restart the Synchronization Queue DM.

Configuring a Timeout for Database Connection Attempts

You can configure how often the Synchronization Queue DM attempts to connect to the database that contains the Synchronization Queue DM queue.

To configure the timeout period for **dm_ifw_sync**:

1. Open the *BRM_home*/**sys/dm_ifw_sync/pin.conf** file in a text editor.
2. The timeout period in milliseconds. **dm_ifw_sync** will wait for the specified number of milliseconds for a response from the database. If the database does not respond within this time, **dm_ifw_sync** throws an error and aborts the transaction. The default is **0**.

   ```
   - dm_ifw_sync database_request_timeout_duration duration
   ```

3. Save and close the file.

To configure the timeout period for **dm_aq**:

1. Open the *BRM_home*/**sys/dm_aq/pin.conf** file in a text editor.
2. The timeout period in milliseconds. **dm_aq** will wait for the specified number of milliseconds for a response from the database. If the database does not respond within this time, **dm_aq** throws an error and aborts the transaction. The default is **0**.

   ```
   - dm_aq database_request_timeout_duration duration
   ```

3. Save and close the file.

Disconnecting and Reconnecting the Synchronization Queue DM to the Queue

You can prevent the Synchronization Queue DM from enqueuing business events when you tune or shut down the queuing database by using the **pin_ctl** utility.

To *disconnect* from the queue, use the following command:

```
pin_ctl stop dm_aq
```

To *reconnect* to the database queue and begin enqueuing business events, use the following command:

```
pin_ctl start dm_aq
```

Troubleshooting the Synchronization Queue DM

If an error occurs during a Synchronization Queue DM operation, check the Synchronization Queue DM log file (*BRM_home*/**sys/dm_aq/dm_aq.pinlog**) for error codes. For a list of standard errors and codes, see "BRM Error Codes" in *BRM System Administrator's Guide*.

Database Queue Creation Error

To install the Synchronization Queue DM and create a database queue, the **pin** user must have Oracle AQ privileges. If **pin** does not have privileges, you receive the following error when you attempt to install the Synchronization Queue DM or create queues with the **pin_portal_sync_oracle** utility:

```
PLS-00201 identifier 'SYS.DBMS_AQ' must be declared
```

To fix this error:

1. Using SQL*Plus, log in to your database as the SYS user and grant AQ privileges to user **pin**:

```
% sqlplus sys@databaseAlias as sysdba
Enter password: password

SQL> grant execute on dbms_aq to pin;

Grant succeeded.

SQL> grant execute on dbms_aqadm to pin;

Grant succeeded.

SQL> grant select on sys.gv_$aq to pin;

Grant succeeded.

SQL> grant execute on dbms_lock to pin;

Grant succeeded.
```

```
SQL> grant execute on dbms_aqin to pin;

Grant succeeded.

SQL> grant execute on sys.dbms_aqin to pin;

Grant succeeded.
```

2. Reinstall the Synchronization Queue DM or create your queue by running the **pin_portal_sync_oracle** utility manually. See "Manually Creating a Database Queue on an Oracle Database".

3

Exporting Pricing Data in a Batch

This document describes exporting Oracle Communications Billing and Revenue Management (BRM) pricing data for use by external applications.

See also:

- Understanding the Synchronization Queue Data Manager
- Installing and Configuring the Synchronization Queue DM

About Exporting Pricing Data to External CRMs

You can export charge offers, discount offers, and chargeshare offers from BRM to external customer relationship management (CRM) applications in a batch. You might do this for the following reasons:

- Start a CRM integration by sending your existing pricing data to the external CRM.
- Supplement real-time synchronization of pricing data with the external CRM.

You export charge offers, discount offers, and chargeshare offers in a batch by using the "pin_export_price " utility. This utility retrieves the data from the BRM database and uses the Synchronization Queue Data Manager (DM) to publish the data to a Synchronization Queue DM database queue. The external CRM can retrieve the data from the database queue by using processes implemented using Oracle Fusion Middleware and Oracle Application Integration Architecture.

> **Note:**
>
> For more information, see "Using BRM with Oracle Application Integration Architecture" in *BRM Developer's Guide*.

To synchronize data in real time, see "Understanding the Synchronization Queue Data Manager".

About pin_export_price

The **pin_export_price** utility is a standalone multithreaded application (MTA) that synchronizes charge offer, discount offer, and chargeshare offer data between BRM and an external CRM in a batch.

The **pin_export_price** utility performs the following operations:

- Searches for **/product**, **/discount**, and **/sponsorship** objects that meet the criteria set in the utility's parameters.
- Retrieves the data for those objects.

- Generates the following notification events, which detail the **/product**, **/discount**, and **/sponsorship** objects that are being exported:
  - **/event/notification/price/discounts/modify**
  - **/event/notification/price/products/modify**
  - **/event/notification/price/sponsorships/modify**

The data is sent to the Payload Generator External Module (EM), which then sends the data in the form of business events to the Synchronization Queue DM. The Synchronization Queue DM sends the business events to the Synchronization Queue DM database queue.

For more information about:

- Business events, see "About the EAI Framework" in *BRM Installation Guide*.
- Synchronization Queue DM, see "Understanding the Synchronization Queue Data Manager".
- Synchronization Queue DM database queues, see "About the Synchronization Queue DM Database Queue".

Setting Up BRM to Export Pricing Data

To set up BRM to export pricing data in a batch, you must:

- Install and configure the Synchronization Queue Data Manager and start the Synchronization Queue DM (**dm_aq**). See "Installing and Configuring the Synchronization Queue DM".
- Edit the **pin_export_price** configuration file (*BRM_home***/apps/pin_export_price/pin.conf**).

  This configuration file has the standard Connection Manager (CM) connection and MTA entries. See:
  - "About Configuration Files" in *BRM System Administrator's Guide*.
  - "Configuring Your Multithreaded Application" in *BRM Developer's Guide*.

> **Note:**
>
> This configuration file must be present in the location from which you run the utility.

Using pin_export_price to Export Pricing Data

You can use the **pin_export_price** utility to export the following from BRM to the Synchronization Queue DM database queue:

- All pricing data (charge offers, discount offers, and chargeshare offers). See "Exporting All Pricing Data".
- Pricing data based on the object type. See "Exporting Pricing Data Based on the Object Type".

- Pricing data based on the modification time. See "Exporting Pricing Data Based on the Modification Time".
- Pricing data based on the service type. See "Exporting Pricing Data Based on the Service Type".

For a list of **pin_export_price** parameters and the syntax for using them, see **"pin_export_price "**.

Exporting All Pricing Data

To export all pricing data (charge offers, discount offers, and chargeshare offers) in your system as a batch, go to the *BRM_home***/apps/pin_export_price** directory and enter the following command:

```
pin_export_price
```

Exporting Pricing Data Based on the Object Type

To export pricing data based on the object type, go to the *BRM_home***/apps/pin_export_price** directory and enter one of these commands:

- To export only **/product** objects:

  ```
  pin_export_price -p
  ```

- To export only **/discount** objects:

  ```
  pin_export_price -d
  ```

- To export only **/sponsorship** objects:

  ```
  pin_export_price -s
  ```

Exporting Pricing Data Based on the Modification Time

To export all or a subset of the pricing data based on the modification time, go to the *BRM_home***/apps/pin_export_price** directory and enter the following command:

```
pin_export_price [-p | -d | -s] -t timestamp
```

where *timestamp* specifies to retrieve only pricing data that was modified on or after the specified timestamp. You specify the timestamp by using the ISO-8601 standard format: *YYYY-MM-DDThh*:*mm*:*ss*[**+|-**]*TZD*. If you do not specify *TZD*, the utility uses the server's local time zone as the default. If your server and database are in different time zones, you must specify the time zone.

For example, the following command exports all charge offers that were modified after 12:00 PM on April 15, 2007, in the UTC-9 time zone:

```
pin_export_price -p -t 2007-04-15T12:00:00-09:00
```

Exporting Pricing Data Based on the Service Type

To export **/product** or **/discount** objects based on the service type, go to the *BRM_home***/apps/pin_export_price** directory and enter the following command:

```
pin_export_price {-p | -d} -S ServiceType1 [, ServiceType2 ...]
```

where *ServiceTypeN* specifies the service's storable class name, such as **/service/telco/gsm/data**. You can specify multiple service types by using a comma (**,**) as a delimiter.

> **Note:**
>
> You cannot export **/sponsorship** objects based on the service type. The utility fails if you use the **-S** *ServiceTypeN* parameter with the **-s** parameter.

> **Note:**
>
> The utility does not validate the name of the service type passed in with the **-S** parameter. If an invalid service type name is passed, the application does not return anything and does not report an error.

For example, the following command exports all charge offers that are associated with the **/service/telco/gsm/telephony** and **/service/telco/gsm/data** service types:

```
pin_export_price -p -S /service/telco/gsm/telephony, /service/telco/gsm/data
```

4

Synchronization Queue Data Manager Utilities

This document provides reference information for Oracle Communications Billing and Revenue Management (BRM) Synchronization Queue Data Manager (DM) utilities.

Topics in this document:

pin_export_price

Use this utility to export in a batch some or all charge offers, discount offers, and chargeshare offers in your Oracle Communications Billing and Revenue Management (BRM) system to an external application.

This utility collects data about charge offers, discount offers, and chargeshare offers according to the criteria specified by the command-line parameters. The utility then triggers notification events that send the pricing data to the Synchronization Queue Data Manager (DM) database queue via the Payload Generator External Module (EM) and Synchronization Queue DM. The data is available to the external application from the Synchronization Queue DM database queue.

For more information on exporting pricing data in a batch, see "Exporting Pricing Data in a Batch".

> **Note:**
>
> - The **pin_export_price** utility needs the configuration file in *BRM_home***/apps/pin_export_price**. This configuration file must be present in the directory from which you run the utility.
> - Because **pin_export_price** is based on the BRM multithreaded application (MTA) framework, the configuration file also requires some performance-related configuration entries. For more information, see "Configuring Your Multithreaded Application" in *BRM Developer's Guide*.
> - The Synchronization Queue DM (**dm_aq**) must be running to use **pin_export_price**. See "Installing and Configuring the Synchronization Queue DM".

Location

*BRM_home***/bin**

Syntax

```
pin_export_price [-p | -d | -s] [-S ServiceType1, ServiceType2,...] [-t
timestamp] [-help]
```

Parameters

-p
Collects data about charge offers (**/product** objects) only. If you do not specify **-p**, **-d**, or **-s**, the utility collects data about charge offers, discount offers, and chargeshare offers.

-d
Collects data about discount offers (**/discount** objects) only. If you do not specify **-p**, **-d**, or **-s**, the utility collects data about charge offers, discount offers, and chargeshare offers.

-s
Collects data about chargeshare offers (**/sponsorship** objects) only. If you do not specify **-p**, **-d**, or **-s**, the utility collects data about charge offers, discount offers, and chargeshare offers.

-S *ServiceType1*, *ServiceType2*,...
Collects data about charge offers and discount offers for the specified service types. In place of *ServiceType1* and *ServiceType2*, use the full name of a service type, such as **/service/telco/gsm/telephony**. Delimit multiple service types by using a comma (**,**). Wildcards are not supported in service type names.
For example:

```
-S /service/telco/gsm/telephony, /service/telco/gsm/data
```

> **Note:**
>
> This option can be used with the **-d** or **-p** parameter only; the utility fails if it is used with the **-s** parameter.

> **Note:**
>
> The utility does not validate the name of the service type passed in with the parameter. If an invalid service type is passed, the utility does not return anything and does not report an error.

-t *timestamp*
Collects data about the charge offers, discount offers, and chargeshare offers modified on or after the specified timestamp. You specify the timestamp by using the ISO-8601 standard. The following formats in Table 4-1 are supported:

Format	Time zone
YYYY	Local time of system used to run **pin_export_price**.

Format	Time zone
YYYY-MM	Local time of system used to run **pin_export_price**.
YYYY-MM-DD	Local time of system used to run **pin_export_price**.
*YYYY-MM-DD*T*hh*:*mm*Z	UTC
*YYYY-MM-DD*T*hh*:*mm*:*ss*Z	UTC
*YYYY-MM-DD*T*hh*:*mm*[+|-]*TZD*	*TZD* is the time zone relative to UTC. You can use a negative or positive offset from 00:00 to 12:00; for example, **-05:00** or **+10:00**.
*YYYY-MM-DD*T*hh*:*mm*:*ss*[+|-]*TZD*	*TZD* is the time zone relative to UTC. You can use a negative or positive offset from 00:00 to 12:00; for example, **-05:00** or **+10:00**.

For example:

```
-t 2005-07-16T19:20:30+01:00
```

-help
Displays the syntax and parameters for this utility.

Results

To check the results of running this utility, look in the **pin_export_price.pinlog** log file for error messages. You specify the directory in which the log file is created in the utility's configuration file (**pin.conf**).

pin_portal_sync_oracle

Use this utility to create, drop, and monitor Oracle Communications Billing and Revenue Management (BRM) Synchronization Queue Data Manager (DM) queues in your Oracle database.

The Synchronization Queue DM sends BRM business events to these queues so that BRM pricing data can be synchronized with data on external systems. For information, see "Understanding the Synchronization Queue Data Manager".

Location

*BRM_home***/apps/pin_aq**

Syntax Overview

The following actions are supported for Oracle databases:

- Syntax for Creating a Queue
- Syntax for Dropping a Queue
- Syntax for Generating a Report
- Syntax for Testing a Queue
- Syntax for Listing Queues

- Syntax for Getting Help

Syntax for Creating a Queue

Creates a Synchronization Queue DM queue, queue table, and database package in your database. The database package contains stored procedures for queuing, dequeuing, and purging business events.

```
pin_portal_sync_oracle.pl create [-l username/password@databaseAlias]
                                 [-q queue_name -t queue_table]
                                 [-s storage_clause]
                                 [-r retention_time]
```

Parameters for Creating a Queue

-l *username*/*password*@*databaseAlias*
Specifies how to connect to the database. For example:

```
pin_portal_sync_oracle.pl create -l pin/password@pindb.example.com
```

If you omit this parameter, the utility prompts you for this information.

-q *queue_name* -t *queue_table*
Specifies the queue name and queue table name.
If you omit these parameters, the utility automatically creates a queue named AQ_QUEUE and a queue table named AQ_SYNC.

-s *storage_clause*
Specifies the storage settings for the queue table.
If you omit this parameter, the storage settings are set by the **storage_clause** parameter in the *BRM_home*/**apps/pin_aq/create_sync_queue.conf** file. For information, see "Configuring the Queue Location".
For example:

```
pin_portal_sync_oracle.pl create -s "tablespace PIN00 initrans 5 storage
(initial 200k next 200k maxextents unlimited pctincrease 0 )"
```

-r *retention_time*
Specifies the amount of time, in seconds, until processed events are removed from the database queue.
If you omit this parameter, the retention time is set by the **retention_time** parameter in the *BRM_home*/**apps/pin_aq/create_sync_queue.conf** file. For information, see "Configuring How Often Processed Events are Removed from the Queue".

Syntax for Dropping a Queue

Drops the specified queue and its associated queue table from your database.

> **Note:**
>
> This command also drops the Synchronization Queue DM database package, which contains stored procedures for queuing, dequeuing, and purging events.

pin_portal_sync_oracle.pl drop [**-q** *queue_name*] [**-l** *username*/*password*@*databaseAlias*]

Parameters for Dropping a Queue

-q *queue_name*
Specifies the name of the queue to drop.
If you omit this option, the utility drops the default queue, PORTAL_SYNC_QUEUE.

-l *username/password@databaseAlias*
Specifies how to connect to the database.
If you omit this option, the utility prompts you for this information.

Syntax for Generating a Report

Generates a report that displays the state of each event in the Synchronization Queue DM queue.

pin_portal_sync_oracle.pl report -r summary|detail [**-q** *queue_name*] [**-l** *username*/*password*@*databaseAlias*]

Parameters for Generating a Report

-r summary | detail
Generates the specified type of report.
-r summary generates a report that summarizes the number of events in each state. Events can be in the following states:

State	Description
READY	The event has not been dequeued or processed by the external system.
PROCESSED	The event was dequeued and processed by the external system.

-r detail generates a report that details the ID, event state, queuing time, and dequeuing time for each event.

-q *queue_name*
Specifies the queue name.
If you omit this parameter, the utility generates a report for the default queue, PORTAL_SYNC_QUEUE.

-l *username/password@databaseAlias*
Specifies how to connect to the database.
If you omit this parameter, the utility prompts you for this information.

Syntax for Testing a Queue

Tests the specified queue by attempting to queue and dequeue 20 test events. You run this command to test if a newly created queue functions properly.

> **Note:**
>
> You need to test a queue only after it is first created.

```
pin_portal_sync_oracle.pl test [-q queue_name] [-l username/
password@databaseAlias]
```

Parameters for Testing a Queue

-q *queue_name*
Specifies the queue name.
If you omit this parameter, the utility tests the default queue, PORTAL_SYNC_QUEUE, and default queue table, PORTAL_SYNC.

-l *username/password@databaseAlias*
Specifies how to connect to the database.
If you omit this parameter, the utility prompts you for this information.

Syntax for Listing Queues

Lists all queues in the current user's database.

```
pin_portal_sync_oracle.pl list [-l username/password@databaseAlias]
```

Parameters for Listing Queues

-l *username/password@databaseAlias*
Specifies how to connect to the database.
If you omit this parameter, the utility prompts you for this information.

Syntax for Getting Help

Displays the syntax and parameters for this utility.

```
pin_portal_sync_oracle.pl help
```

Results

The **pin_portal_sync_oracle** utility notifies you when it runs successfully. Otherwise, look in the **default.pinlog** file for errors. This file is either in the directory from which the utility was started or in a directory specified in the utility configuration file.