Oracle® Communications Billing and Revenue Management

Moving Accounts Between Database Schemas

Release 15.2

G35872-01

January 2026

Oracle Communications Billing and Revenue Management Moving Accounts Between Database Schemas, Release 15.2

G35872-01

Contents

4 Installing and Configuring BRM for Account Migration

5 Using Account Migration Manager

About This Content

This book describes how to move accounts between BRM database schemas. You move accounts to balance the number of accounts between schemas.

Audience

This book is for system administrators and operations personnel.

1

Understanding Account Migration

Learn about account migration which is the process of transferring data associated with selected accounts from the source database schema to a destination schema. In Oracle Communications Billing and Revenue Management (BRM) you use Account Migration Manager (AMM) to migrate accounts.

Topics in this document:

About Account Migration

Account migration is the process of transferring data associated with selected accounts from the data's source database schema to a destination schema.

You use the AMM application to perform all account migration tasks. You provide information on the accounts to migrate and how to access the source and destination database schemas. See "About the AMM Application".

You migrate account objects to redistribute the data in the following situations:

- One BRM database schema contains significantly more or fewer accounts than other BRM database schemas (for example, when you add a schema to an existing multischema system).
- The number of events per account is significantly greater in one schema than another.
- The time it takes to complete a billing run becomes erratic.

You achieve optimal migration performance and the smallest impact on your operations if you schedule migrations for maintenance windows. If you need to migrate a large number of accounts, but your maintenance window is only a couple of hours, you can perform migrations over several days. The AMM software processes jobs in stages, enabling you to pause and resume account migration without affecting database integrity.

For information on scheduling migration, see "Automating Account Migration".

About the AMM Application

AMM is an application that migrates accounts and their associated objects from a source schema in a BRM database to a destination schema in the same database.

The AMM software migrates accounts based on the information you provide in the account search configuration file for that migration task. In this file, you specify a group of accounts to migrate based on specific criteria, such as account creation date or account status. The accounts and associated objects that meet these criteria form a *job*.

Jobs are processed by the AMM software through a queue system, with each job processed in the order received. To improve migration performance, AMM subdivides jobs into *batches* containing a configurable number of accounts.

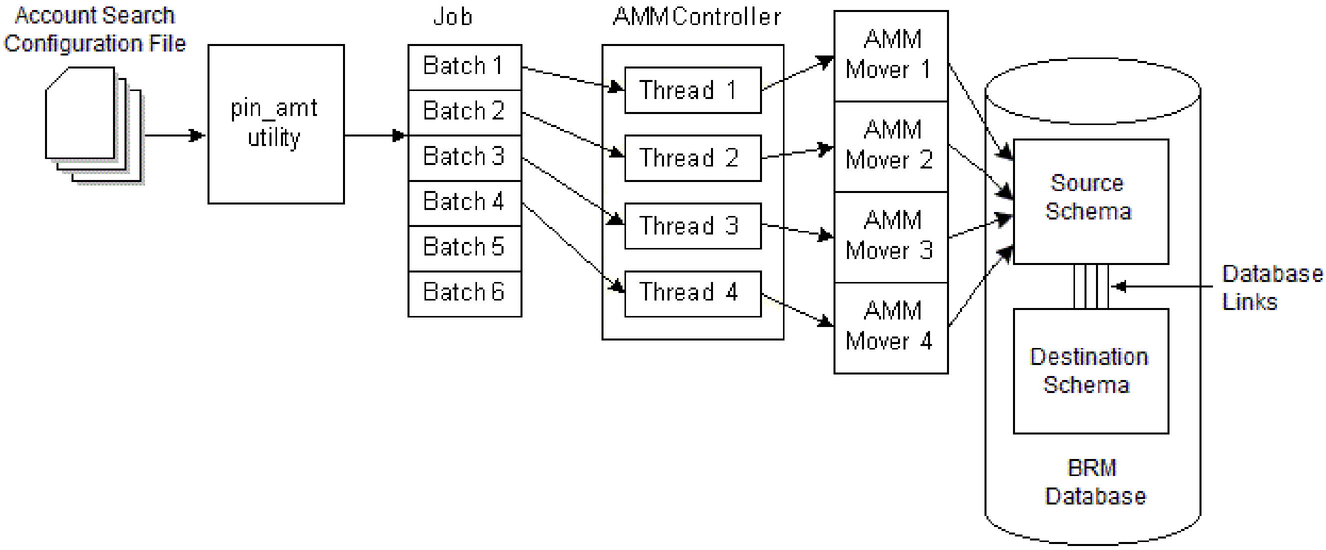
Batches are assigned to a configurable number of threads, which process the batches in parallel. Each batch is migrated in a single distributed transaction, during which activity is prevented on the accounts in the batch. Depending on the success of the batch migration, changes to the database are either committed or rolled back.

The AMM software does not automatically remove migrated objects from the source database schema for performance reasons. Instead, it flags the migrated objects as *invalid* to prevent BRM applications from accessing them. You can use the AMM software to purge invalid objects from the database at any time.

AMM Process Overview

Figure 1-1 shows the AMM process overview.

Figure 1-1 AMM Process Overview



The account migration process can be divided into the following stages:

1. The account search configuration file containing information on the accounts to migrate and access information for the source and destination database schemas is provided as input to the **pin_amt** utility. See "About the Account Search Configuration File".
2. The **pin_amt** utility processes the account search configuration file.
3. Every account to be migrated is set up as an individual job. The jobs are divided into batches and placed in the queue. See "About the pin_amt Utility".
4. The AMM Controller allocates batches to threads and passes batches to the AMM Mover. See "About the AMM Controller".
5. The AMM Mover migrates the batch of accounts by moving the associated data from the source schema to the destination schema. See "About the AMM Mover".

About the Account Search Configuration File

The account search configuration file provides the details on which accounts to migrate, their source and destination storage locations, and the size of each batch.

You can also migrate accounts based on custom criteria. See "Creating Custom Account Search Criteria".

About the pin_amt Utility

The **pin_amt** utility is a standalone utility that generates account migration jobs for the AMM Controller to process. This utility can perform all its functions, such as finding accounts to migrate and submitting and enabling jobs, whether the AMM Controller is online or offline.

You use the **pin_amt** utility to perform the following tasks:

- Start, stop, resume, and monitor the AMM Controller
- Find all accounts in the source database schema that meet your search criteria
- Enable account migration jobs in the queue
- Delete jobs from the job management tables
- Purge invalid objects from the source schema

When you submit an account search configuration file, the **pin_amt** utility does the following:

1. Searches the source database schema for all accounts that meet the criteria in the account search configuration file, excluding all accounts that are part of hierarchical and sharing groups.
2. Divides the list of account POIDs into batches.
3. Populates the job management tables in the primary, source, and destination schemas with the list of account POIDs to migrate. See "AMM Job Management Tables".
4. Determines whether group migration is enabled.
   - If group migration is enabled, **pin_amt** proceeds with steps 5 through 10.
   - If group migration is disabled, the account search is complete. The AMM Controller can begin processing the job when the job is enabled in the queue. See "About the AMM Controller".
5. Searches all hierarchical accounts and sharing groups in the source schema for accounts that meet the search criteria.
6. Determines whether to search for and exclude accounts belonging to cross-schema sharing groups from migration.
   - When cross-schema sharing groups are disabled, **pin_amt** does not search for cross-schema sharing group members.
   - When cross-schema sharing groups are enabled, **pin_amt** finds accounts that are members of cross-schema sharing groups and excludes them from migration.

> ⓘ **Note**
>
> AMM performs a check on an account and only its immediate child account. Ensure you perform extra validation to ensure accounts picked by AMM are not part of a cross-schema sharing group.

7. When an account meets the search criteria, **pin_amt** finds all other accounts related to it and organizes them into an account group.
8. Determines whether the size of the account group exceeds the maximum. If it does, AMM excludes the account group from the job.

> ⓘ **Note**
>
> You specify the maximum size of an account group by using the account search configuration file. See "Creating the Account Search Configuration File".

9. Divides all members of one account group into batches. All batches in the group are assigned the same group ID and flagged as containing account group members.
10. Populates the job management tables in the primary, source, and destination schemas with the list of account POIDs to migrate. See "AMM Job Management Tables".

About the AMM Controller

The AMM Controller is a server process that checks the queue for jobs to process. By default, your system contains one AMM Controller, which processes one job at a time. For information about using multiple AMM Controllers, see "About Using Multiple AMM Controllers".

The AMM Controller processes group member and nongroup member batches in different ways:

- How AMM Controller Processes Batches with Nongroup Members
- How AMM Controller Processes Batches with Account Group Members

How AMM Controller Processes Batches with Nongroup Members

When processing batches that contain nongroup members, the AMM Controller does the following:

1. Assigns batches to threads, which run in parallel. Each thread processes one batch at a time. If there are more batches than threads, each thread must process multiple batches in sequence. You use a configuration file to configure the number of AMM Controller threads. See "Connecting AMM to Your Database Schemas".
2. Changes the batch status to IN_PROGRESS. See "About Batch Status Flags".
3. Passes the job ID, batch number, and schema qualifications to the AMM Mover on the destination database schema for processing. See "About the AMM Mover".
4. Determines whether the AMM Mover successfully migrated accounts.
   - If migration is successful, the AMM Controller commits the changes to all database schemas and changes the batch status to FINISHED. See "About Batch Status Flags".
   - If migration fails, the AMM Controller rolls back the changes and updates the batch status to FAILED. See "About Batch Status Flags".

How AMM Controller Processes Batches with Account Group Members

When processing batches that contain account group members, the AMM Controller does the following:

1. Changes the account group status to GROUP_DISABLING. See "About Group Status Flags".
2. Locks the appropriate base table records in the source database schema so that applications cannot access group member accounts during migration.
3. Marks all account group members in the source schema as invalid.
4. Determines whether all account group members were disabled in the source schema.
   - If all accounts were successfully disabled, the AMM Controller changes the account group status to GROUP_READY. See "About Group Status Flags".
   - If any accounts were not disabled, the AMM Controller changes the account group status to FAILED. See "About Group Status Flags".
5. Changes the account group status to GROUP_IN_PROGRESS.
6. Passes individual batches in the account group to the AMM Mover for processing.
   a. Assigns an individual batch in the group to a thread.
   b. Changes the batch status to IN_PROGRESS.
   c. Passes the job ID, batch number, and schema qualifications to the AMM Mover. See "About the AMM Mover".
   d. Determines whether the AMM Mover successfully migrated the batch and sets the batch status to FINISHED or FAILED.
7. Determines whether all batches in the account group migrated successfully.
   - If all batches migrated successfully, the AMM Controller enables all account group members in the destination database schema and changes the account group status to GROUP_FINISHED. See "About Group Status Flags".
   - If any batch failed to migrate, the AMM Controller changes the account group status to GROUP_FAILED. See "About Group Status Flags".

> **Note**
>
> When an account group fails to migrate, all its accounts remain disabled in the source and destination schemas. You must fix the error and migrate the job again before your BRM system can access the accounts.

About the AMM Mover

The AMM Mover is the process that actually moves accounts from one database schema to another. Each schema contains at least one AMM Mover.

Note

The AMM Mover performs the following functions regardless of whether a batch contains group-member or nongroup member accounts.

When the AMM Mover receives a batch, it does the following:

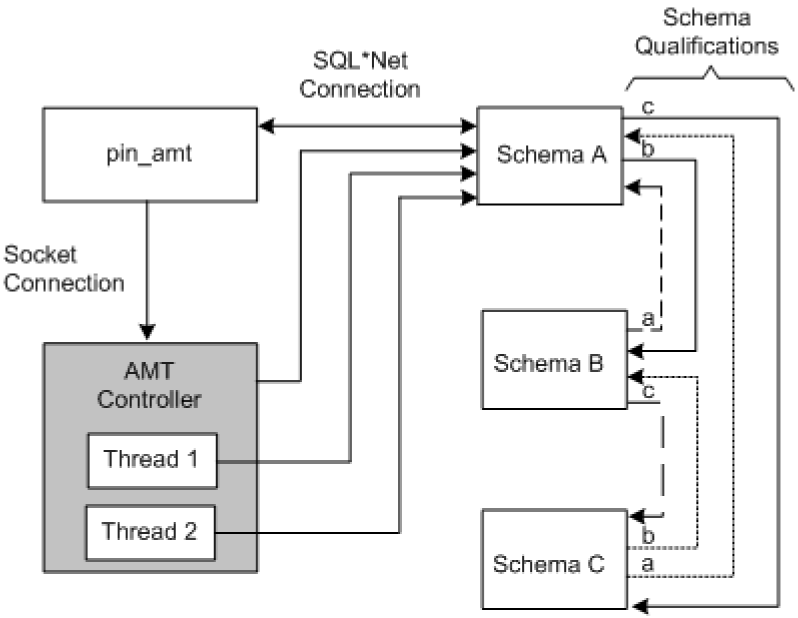
1. Locks the appropriate base table records in the source database schema so that applications cannot access accounts during migration.
2. Migrates the objects for an account batch from the source schema to the destination schema in a single distributed transaction. See "About Distributed Transactions".
3. Marks all migrated objects in the source schema as invalid.
4. Updates the account POIDs in the uniqueness table to reflect the account's new location. For example, if an account is migrated from schema **0.0.0.1** to schema **0.0.0.2**, the account POID changes from **0.0.0.1 /account 2286 0** to **0.0.0.2 /account 2286 0**.

About Distributed Transactions

The AMM software migrates each batch of accounts as a single distributed transaction by using schema qualifications. This means that changes can be made to the primary, source, and destination database schemas and then committed or rolled back in one transaction, ensuring the integrity of the database.

Figure 1-2 shows the schema qualifications for a multischema system with three database schemas, one AMM Controller, and two threads:

Figure 1-2 AMM Database Schema Connections



About Using Multiple AMM Controllers

Note

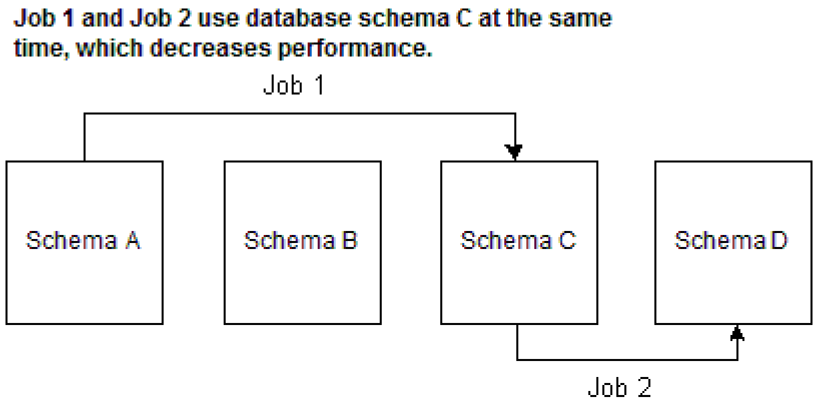
Implementing multiple AMM Controllers is for advanced users only. Use multiple AMM Controllers only if you understand the impact on migration performance.

Using multiple AMM Controllers enables you to process multiple account migration jobs in parallel. However, you receive performance improvements only in the following situations:

- Your system contains more than three database schemas.
- No two migration jobs use the same schema at the same time.

When multiple jobs use the same schema, as shown in Figure 1-3, migration performance degrades significantly.

Figure 1-3 Concurrent Database Use Performance Degradation


Job 1 and Job 2 use database schema C at the same time, which decreases performance.
Job 1
Schema A
Schema B
Schema C
Schema D
Job 2

For more information, contact your Oracle BRM representative.

Account Migration Performance

Account migration is resource intensive and can overload your BRM system.

Signs that your system is overloaded during account migration:

- Batch processing times steadily increase, without returning to their initial processing times.
- The AMM software is processing fewer than five accounts per second.
- You receive a distributed transaction time-out error (Oracle error 2049).
- There are a high number of waits for undo segment extension and latch free operations.

If your system exhibits any of these signs, you need to tune your Oracle database. For guidelines, see "Tuning Your Database for Optimal Account Migration Performance" or contact your Oracle BRM representative.

2

Migrating Hierarchical and Sharing Accounts

Learn how to migrate hierarchical and sharing groups from one database schema to another in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About Migrating Hierarchical and Sharing Accounts
- About Searching for Member and Nongroup Member Accounts
- About Account Groups
- About Migrating Account Groups

About Migrating Hierarchical and Sharing Accounts

You can configure AMM to migrate hierarchical account groups and charge and discount sharing groups from a source BRM database schema to a destination schema. In this configuration, AMM does the following:

- Searches for accounts in two phases. See "About Searching for Member and Nongroup Member Accounts".
- Organizes accounts that meet the search criteria by account group. See "About Account Groups".
- Migrates entire account groups. See "About Migrating Account Groups".

By default, group migration is disabled. You specify whether to migrate account groups by using the **migration_mode** entry in the account search file (*BRM_home***/apps/amt/account_search.cfg**). See "Creating the Account Search Configuration File".

> **Note**
>
> When you enable group migration, you must perform extra verification steps to prevent accounts from being severed from their associated account group. See "Checking Account Group Details".

About Searching for Member and Nongroup Member Accounts

When you enable group migration, AMM searches for accounts as follows:

1. **AMM searches for nongroup member accounts only.** That is, AMM excludes all hierarchical and sharing group accounts from the account search. Accounts meeting the search criteria are divided into batches, and each batch is flagged as containing nongroup member accounts only.
2. **AMM searches for accounts belonging to hierarchical and sharing groups only.** If an account meets the search criteria, AMM finds all other account members that are related to

the account. These accounts are organized into an *account group*. See "About Account Groups".

Each account group is divided into batches, which are assigned an account group ID and flagged as containing account group members.

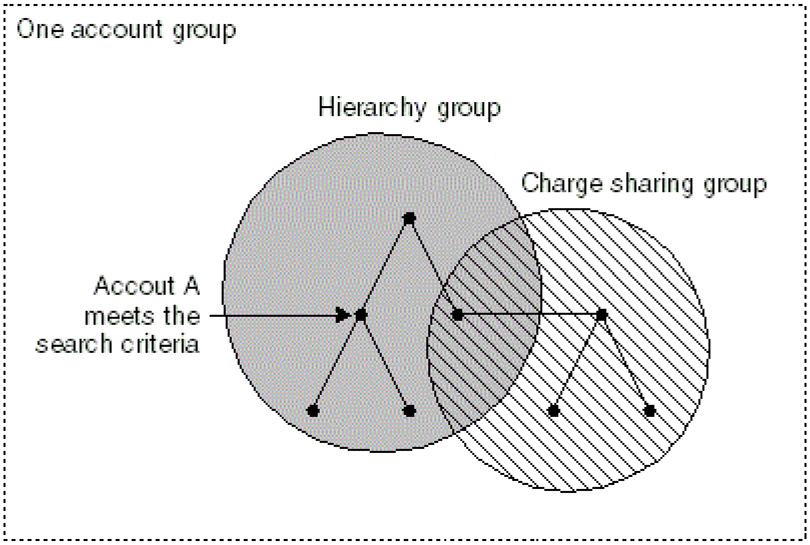
> **Note**
>
> You can configure AMM to exclude from migration any accounts belonging to a cross-schema sharing group, which means that members can reside in multiple database schemas. To do so, set the **cross_schema_group** parameter to **true** in the account search file. See "Creating the Account Search Configuration File".

All accounts meeting the search criteria, both group member and nongroup member accounts, still form one job.

About Account Groups

An account group consists of all account members that are related to a specific account. When AMM finds an account that meets the search criteria, it finds the parent account and all other child accounts in the group. If one of the accounts is also a member of another group, it finds all members of the other group as well.

Figure 2-1 One Account Group



For example, in Figure 2-1, account A meets the search criteria. Because account A is a child in a hierarchical account group, AMM finds the parent account and all child accounts in that group. Because one hierarchical account member is also a member of a charge sharing group, AMM finds all accounts in the charge sharing group as well. In this example, the account group consists of all accounts in the hierarchical account group and the charge sharing group.

About Migrating Account Groups

AMM migrates account group member and nongroup member batches in different ways:

- **Batches containing nongroup members in one transaction.** During migration, AMM disables all accounts that belong to a single batch.
  - If the batch migrates successfully, AMM commits the changes to all database schemas and enables the accounts in the *destination* schema.
  - If the batch fails to migrate, AMM rolls back the changes and re-enables the accounts in the *source* database schema.
- **Batches containing account group members by account group ID.** AMM disables all accounts that belong to an account group before migration begins. After migration starts, AMM monitors whether all batches for the account group migrate successfully.
  - If all batches migrate successfully, AMM commits the changes to all database schemas and enables the accounts in the *destination* schema.
  - If even one batch in the group fails, AMM leaves all account group members disabled in both source and destination schemas. You must fix the error and use AMM to reprocess the job before your BRM system can access the accounts.

3 Account Migration Restrictions

Learn about the account migration restrictions for using Oracle Communications Billing and Revenue Management (BRM) Account Migration Manager (AMM).

Topics in this document:

Account Activity Prevented during Account Migration

To prevent applications from accessing or modifying accounts that are being migrated, AMM locks the accounts in Oracle. Only one batch of accounts per thread is locked at a time and only while the accounts are being physically migrated.

> **Note**
>
> When migrating account groups, AMM locks all accounts in the account group before migration begins.

If an application attempts to access a locked account, the Oracle Data Manager (DM) returns a PIN_ERR_INVALID_OBJECT error.

Do Not Rerate Events during Account Migration

Because the AMM software may suspend some events that you want to rerate, you *must not* rerate events during account migration.

Do Not Alter Account Group Members

AMM checks account group relationships only when you first create a job, and does not re-verify relationships during the migration process. Therefore, once an account group is included in a migration job, you:

- *Must not* add members to the account group

- *Must not* modify relationships between account group members

If you need to alter an account group *after* it is included in a job but *before* the job completes migration, you must:

1. Delete the migration job.
2. Modify the account group.
3. Re-create the account migration job.

Migration Prevented during Account Activity

AMM does not migrate accounts while they are being accessed or modified by BRM or another application. For best performance, stop all account activity before you migrate accounts.

If you cannot restrict all access to the accounts in your database schemas, AMM can still process account migration jobs. However, AMM does not migrate any batch that contains active accounts. You can check for failed batches and resubmit them for migration once account activity stops.

Unique POIDs Required across All Database Schemas

The AMM software can migrate only accounts that have a unique POID. Starting with BRM Release 6.2 ServicePack 1, the multischema software automatically creates unique POIDs across all database schemas in your system.

If your database contains accounts created both prior to and after Release 6.2 ServicePack 1 was installed, AMM migrates only those accounts created *after* 6.2 ServicePack 1 was installed. For information on how to migrate accounts created prior to 6.2 ServicePack 1, contact your Oracle BRM representative.

> **Note**
>
> If your BRM system uses a custom POID generation scheme, make sure the sequence number generation algorithm creates unique POIDs across all of your database schemas.

Transient Objects Are Not Migrated

While the **pin_amt** utility migrates account objects, it does not migrate any transient objects.

Some Client Applications May Fail during Account Migration

BRM client applications may generate error messages and fail to commit changes to the database during account migration. For example, if a CSR opens an account in Billing Care just prior to the account being migrated to another database schema, an error might result.

In that case, the CSR must restart Billing Care and access the account again so it retrieves the account's new location.

Note

If your system contains custom client applications that connect to the BRM database and search accounts based on POID, you must modify your application. See "Modifying Custom Client Applications for AMM".

AMM Does Not Support LDAP Manager

Currently, you cannot use AMM to migrate accounts if your BRM system includes LDAP Manager.

Using BRM Reports after Migration

To use BRM Reports after you migrate accounts, you must use BRM Reports Release 6.2 ServicePack 1 or later. If you use an earlier version of BRM Reports with AMM, your reports will retrieve and process duplicate data from your source and destination database schemas.

For example, if an account object is migrated from schema **0.0.0.1** to schema **0.0.0.2**, earlier versions of BRM Reports retrieve the account object from both schemas while BRM Reports 6.2 ServicePack 1 and later retrieve the account object only from schema **0.0.0.2**.

For information on how to modify custom reports to work with AMM, see "Modifying Custom BRM Reports for AMM".

4

Installing and Configuring BRM for Account Migration

Learn how to install and configure software required for account migration with the Oracle Billing and Revenue Management (BRM).

Topics in this document:

- System Requirements
- Installing AMM
- Configuring AMM for Additional Database Schemas
- Configuring AMM for New Custom Tables
- Tuning Your Database for Optimal Account Migration Performance

System Requirements

The system requirements for account migration consists of the following:

- General Software Requirements
- Software Requirements for Account Migration Manager

General Software Requirements

Before installing AMM, you must install:

- The compatible version of the **Oracle Database software**.
- PERL libraries and the JRE required for installing BRM components.
- **Multidatabase Manager**. See “Installing a Multischema System" in *BRM Installation Guide*.

For information about the compatible software versions for your release, see *BRM Compatibility Matrix.*

Software Requirements for Account Migration Manager

AMM is available for Oracle databases and the Linux operating system. For information on disk space requirements for the Linux operating system, see “Disk Space Requirements" in *BRM Installation Guide*.

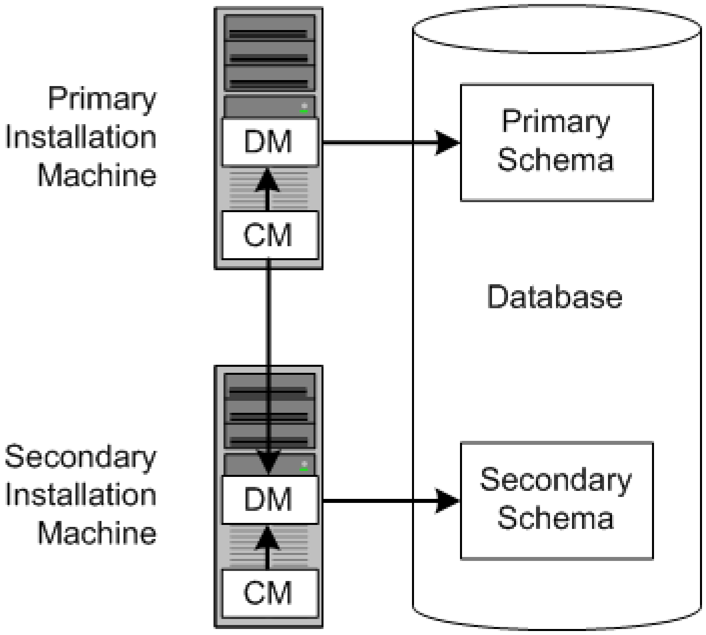
If you plan to use AMM software for account migration, you must also install:

- Java Development Kit

Installing AMM

The instructions in this section assume that you have two BRM installation machines and two database schemas in your multischema environment as shown in Figure 4-1.

Figure 4-1 Installing AMM



Installing AMM involves the following general steps:

1. Configuring All Primary and Secondary Database Schemas
2. Installing the AMM Software on the Primary Installation Machine
3. Configuring Your Oracle DM to Check for Invalid Objects
4. Connecting AMM to Your Database Schemas

Configuring All Primary and Secondary Database Schemas

Before you can install AMM, you must configure your primary and secondary database schemas for account migration.

First, ensure that you assigned unique database instance names to each schema in your system:

1. Open the **tnsnames.ora** file in a text editor.
2. If necessary, assign a unique database instance name to each schema. For example, for the first schema:

   ```
   Alias1 = (DESCRIPTION =  (ADDRESS = (PROTOCOL= TCP)(Host=DatabaseHost1Name)(Port=
   1521))
   (CONNECT_DATA = (SID = DatabaseSID1)) )
   ```

   For the second schema:

```
Alias2 = (DESCRIPTION =  (ADDRESS = (PROTOCOL= TCP)(Host=DatabaseHost2Name)(Port=
1521))
(CONNECT_DATA = (SID = DatabaseSID2)) )
```

3. Save and close the file.

Then perform the following steps on *each* database schema in your multischema system:

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

   ```
   % sqlplus system/password@databaseAlias

   SQL> grant create database link to pin;

   Grant succeeded.
   ```

2. Verify that JServer is installed on your system:

   ```
   SQL> SELECT object_name, object_type FROM all_objects WHERE
        object_type = 'PACKAGE' and object_name = 'DBMS_JAVA';
   ```

   If JServer is installed on your system, you receive the following:

   ```
   OBJECT_NAME                OBJECT_TYPE
   ------------------------   -----------
   DBMS_JAVA                  PACKAGE
   ```

   If JServer is not installed, you receive the following:

   ```
   no rows selected
   ```

3. Install JServer if it is not already installed on your system:

   a. Add the following entry to the Oracle **init***SID***.ora** file (**$ORACLE_HOME/dbs/init***SID***.ora**) for the schema's database instance:

      ```
      java_pool_size=20971520
      ```

   b. Restart Oracle so that the schema's database instance is initialized with your changes.

   c. Install JServer manually by running the Oracle **initjvm** script:

      ```
      % sqlplus sys/password@databaseAlias
      SQL> @$ORACLE_HOME/javavm/install/initjvm.sql
      ```

      For information, see your Oracle documentation.

4. Modify the entries listed in Table 4-1 in the Oracle **init***SID***.ora** file (**$ORACLE_HOME/dbs/init***SID***.ora**) of the schema's database instance:

   Table 4-1 initSID.ora Entries

Entry	Description
global_names	Specifies whether a database link is required to have the same name as the database schema to which it connects. Set this to **False**.
utl_file_dir	Specifies the location of the Oracle **utl_file**. Set this to a writable directory for the user **oracle**.

5. Restart Oracle so that the schema's database instance is initialized with your changes.

Installing the AMM Software on the Primary Installation Machine

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

To install AMM, perform the following steps on the primary installation machine:

1. Install AMM. See “Installing Individual BRM Components" in *BRM Installation Guide*.
2. Verify that the *BRM_home***/setup/scripts/pin_multidb.conf** file contains accurate information about each database schema in your system. The **pin_amt_install** script uses the information in this file to set up your AMM environment.
3. For optimal performance, store your AMM job management tables and indexes in their own physical tablespaces. Map the following entries to four separate physical tablespaces by modifying the *BRM_home***/setup/scripts/pin_tables.values** file:

   ```
   $PIN_CONF_TBLSPACE14
   $PIN_CONF_TBLSPACE15
   $PIN_CONF_TBLSPACEX16
   $PIN_CONF_TBLSPACEX17
   ```

   See "Database Configuration and Tuning" in *BRM Installation Guide*.
4. Log in as user **pin**, go to the *BRM_home***/setup/scripts** directory, and run the **pin_amt_install** script:

   ```
   # su - pin
   % cd BRM_home/setup/scripts
   % perl pin_amt_install.pl
   ```
5. Verify that installation was successful by checking the AMM installation log file (*BRM_home***/setup/scripts/pin_amt_install.log**).
6. Restart the BRM server.

Configuring Your Oracle DM to Check for Invalid Objects

During account migration, the AMM software marks all migrated objects in the source database schema as invalid. To prevent BRM applications from accessing these objects, you must configure your Oracle Data Manager (DM) to check for invalid objects during account searches.

> **Note**
>
> If you do not make this change, BRM applications retrieve duplicate data from your source and destination database schemas. For example, if an account is migrated from schema **0.0.0.1** to schema **0.0.0.2**, the application retrieves the account object from both schemas.

To configure your system to check for invalid objects, perform the following on *every* machine containing an Oracle DM:

1. Add the following line to the *BRM_home***/sys/dm_oracle/pin.conf** file:

   ```
   - dm dm_nul_poid_db_chk 1
   ```

2. Restart **dm_oracle** so that your Oracle DM is initialized with your changes.

Connecting AMM to Your Database Schemas

During installation, the Installer stores the configuration data, such as the database schema connection details and the number and configuration of your AMM Controllers, in the Oracle wallet. AMM retrieves the data from the Oracle wallet. However, if the configuration entries are also stored in the AMM **Infranet.properties** file, AMM retrieves the data from the AMM **Infranet.properties** file. Use the Oracle wallet to provide the information necessary for AMM, such as passwords, to access the data for the accounts in your system.

For more information viewing and storing the configuration entries in the Oracle wallet, see the “About the Oracle Wallet" section in *BRM System Administrator's Guide.*

Configuring Database and AMM Mover Information for Multischema Systems

Verify that the AMM **Infranet.properties** file contains a set of **0.0.0.***x* entries for each database schema in your system. For example, if you have three database schemas in your system, the file should contain a set of entries prefixed by **0.0.0.1**, a set prefixed by **0.0.0.2**, and a set prefixed by **0.0.0.3**. Also, update the BRM wallet in the primary schema to include the user passwords for all the secondary schemas. Example 4-1 shows the entries for a system with two database schemas.

Verify that the **Infranet.properties** file contains information on the AMM Mover associated with each database schema. If a schema contains more than one AMM Mover, make sure to update the **Infranet.properties** file accordingly.

Example 4-1   Example Configuration Information for Multischema Systems

```
#
# primary database schema
#
0.0.0.1_user_name=pin57204
0.0.0.1_instance_name=futt11
0.0.0.1_primary=true
# AMT mover
0.0.0.1_mover_log_file_dir=unknown
0.0.0.1_mover_log_file_flag=N
0.0.0.1_grp_srch_log_file_dir=unknown
0.0.0.1_grp_srch_log_file_flag=N
#
# secondary database schema
#
0.0.0.2_user_name=pin57204m2
0.0.0.2_instance_name=futt11m2
0.0.0.2_primary=false
# AMT mover
0.0.0.2_mover_log_file_dir=unknown
0.0.0.2_mover_log_file_flag=N
0.0.0.2_grp_srch_log_file_dir=unknown
0.0.0.2_grp_srch_log_file_flag=N
#
```

Configuring AMM Controller Definitions

The installer creates a set of **Controller_1** entries for setting up one AMM Controller in the AMM **Infranet.properties** file and populates them with default values, as shown in Example 4-2.

Example 4-2   AMM Controller Definitions

```
# controller definitions
#
controller_1_log_directory=/export/BRM_home/apps/amt
controller_1_port_number=18566
controller_1_server=slc00qhm
controller_1_thread_count=2
controller_1_syslog_priority=7
controller_1_event_generation=false
controller_1_concurrent_job_number=20
controller_1_hold_period=120
controller_1_amt_queue_owner_name=pinq
controller_1_amt_queue_name=ifw_sync_queue_amt
#
```

Access the Oracle wallet and verify that the entries are as required.

If you require more than one AMM Controller, determine the optimal number of AMM Controllers for your system. Make sure that you provide the optimal number of threads for each AMM Controller in your system. See "About Using Multiple AMM Controllers" for more information.

For the additional controllers, create a set of **Controller_2** entries, **Controller_3** entries, and so on in the **Infranet.properties** file.

Specifying Schema Alias Information for Multischema Systems

When you use multischema systems, the database layer of your BRM system consists of one primary schema and one or more secondary schemas in a single database. As a result, the schema alias is the same for both schemas.

Access the Oracle wallet and enter an alias for the secondary schema in the file.

Example 4-3 shows the entries for a multischema system:

Example 4-3   Example Alias Information for Multischema System

```
#
# Connection entries for the Primary Database Schema
0.0.0.1_user_name=pin
0.0.0.1_instance_name=Schema1Alias
0.0.0.1_primary=true
0.0.0.1_mover_log_file_dir=./mover/log
0.0.0.1_mover_log_file_flag=y

# Connection entries for the Secondary Database Schema
0.0.0.2_user_name=pin
0.0.0.2_instance_name=Schema2Alias
0.0.0.2_primary=false
0.0.0.2_mover_log_file_dir=./mover/log
0.0.0.2_mover_log_file_flag=y
```

AMM Infranet.properties File Parameters

Table 4-2 shows the parameters used in defining the AMM **Infranet.properties** file.

Table 4-2 AMM Infranet.properties Values

Entry	Value	Description
0.0.0.*x*_user_name	String	Specifies the Oracle user name for the specified database schema.
0.0.0.*x*_instance_name	String	Specifies the SQL*Net database alias name you assigned in the **tnsnames.ora** file.
0.0.0.*x*_primary	**true** or **false**	Flag that indicates whether the database schema is the primary schema. For the primary schema, set this to **true**. For all secondary schemas, set this to **false**.
0.0.0.*x*_mover_log_file_dir	Path name	Specifies the directory of the AMM Mover log file on the specified database schema. **Important:** This path must match the path specified in the **utl_file_dir** entry of the **init*SID*.ora** file.
0.0.0.*x*_mover_log_file_flag	**Y** or **N**	Specifies whether you want the AMM Mover to create log files on the specified database schema.
0.0.0.*x*_grp_srch_log_file_flag	**Y** or **N**	Specifies whether you want AMM to create a log file for the account group stored procedure. **Note:** The stored procedure finds all account group members related to a specific account.
0.0.0.*x*_grp_srch_log_file_dir	Path name	Specifies the directory for the account group stored procedure log file.
controller_*N*_log_directory	Path name	Specifies the directory in which to create the AMM Controller log file.
controller_*N*_port_number	Integer > 1024	Specifies the TCP/IP port number for the connection between the **pin_amt** utility and the AMM Controller. Each AMM Controller instance requires a unique port number.
controller_*N*_server	String	Specifies the host name of the machine that runs the AMM Controller.
controller_*N*_thread_count	Positive integer	Specifies the number of AMM Controller processing threads. For optimal performance, the number of AMM Controller threads should be 1 to 2 times the number of CPUs on the destination schema that are dedicated to AMM.
controller_*N*_syslog_priority	**1** (low) through **7** (high)	AMM Controller log message priority threshold. Messages with a lower priority are suppressed.
pin_amt_log_directory	Path name	Specifies the path to the **pin_amt** log file.
controller_*N*_event_generation	**true** or **false**	Specifies whether the AMM Controller migrates accounts when ECE is running. This configures AMM to notify ECE when accounts are migrated. The default is **false**.
controller_*N*_concurrent_job_number	Positive integer	Specifies how many jobs the AMM Controller starts concurrently. The default is **20**. **Note:** This entry is required only if the **controller_*N*_event_generation** entry is set to **True**.

Table 4-2 (Cont.) AMM Infranet.properties Values

Entry	Value	Description
controller_*N*_hold_period	Positive integer	Specifies how long the AMM Controller waits, in minutes, before it starts to migrate accounts. This provides time for your ECE to process any events targeted for accounts that are being migrated. The default is **120**. **Note:** This entry is required only if the **controller_*N*_event_generation** entry is set to **True**.
controller_*N*_amt_queue_owner_name	String	Specifies the user that created the acknowledgment queue. **Note:** This entry is required only if the **controller_*N*_event_generation** entry is set to **True**.
controller_*N*_amt_queue_name	String	Specifies the name of the acknowledgment queue. The AMM Controller dequeues ECE acknowledgment events from this queue. **Note:** This entry is required only if the **controller_*N*_event_generation** entry is set to **True**.
infranet.login.type	**1** or **0**	Specifies whether AMM requires a user name and password to log in to the Connection Manager (CM). • **1** specifies that AMM must provide a user name and password. • **0** specifies that AMM uses a “trusted" login that comes through a CM Proxy, for example, and does not require a user name and password in the properties file. **Note:** This entry is required only if the **controller_*N*_event_generation** entry is set to **True**.
publish_migrated_objects	String	Specifies the storable classes whose objects are stored in the MIGRATED_OBJECTS_T cross-reference table. You can list multiple storable classes by using a comma (,) as a delimiter. For example: /service,/billinfo,/payinfo. **Note:** This entry is required only if you integrate AMM with your external application using Oracle Application Integration Architecture (Oracle AIA) in a multischema environment.

Configuring AMM for Additional Database Schemas

You must reconfigure the AMM software whenever you add a database schema to an existing multischema system.

To configure AMM for additional database schemas, perform the following procedure on the primary installation machine:

1. Delete all existing account migration jobs in the queue:

   ```
   % pin_amt -d JobID
   ```

2. Log in as user **pin** and run the **pin_amt_install.pl** script:

   ```
   # su - pin
   % cd BRM_home/setup/scripts
   % perl pin_amt_install.pl
   ```

   This script reinstalls the job management tables on your database schemas.

Configuring AMM for New Custom Tables

AMM migrates data to and from the tables listed in the AMM data dictionary. This list includes all BRM tables and any custom tables that were on your system when you installed AMM. If you add any tables after you install AMM, you must update the AMM data dictionary.

To update the AMM data dictionary, perform the following on your primary installation machine:

1. Log in as user **pin** and go to the *BRM_home***/setup/scripts** directory:

   ```
   % su - pin
   % cd BRM_home/setup/scripts
   ```

2. Run the **pin_amt_install.pl** script with the **-m** parameter:

   ```
   % perl pin_amt_install.pl -m
   ```

   This script updates the AMM data dictionary tables (AMT_META_DATA_T and AMT_POID_TYPE_MAP_T) on all database schemas in your system.

Tuning Your Database for Optimal Account Migration Performance

To tune your database for optimal account migration performance:

- Use cost-based optimization.
- Set the number of Oracle rollback segments to approximately two times the number of AMM Controller threads. Multiple rollback segments enable Oracle to automatically allocate one rollback segment for each transaction.
- Set the **transactions_per_rollback_segment** parameter in the **$ORACLE_HOME/dbs/init***SID***.ora** file to a small number. For best results, set it to 1 or 2.
- Set the initial size for the rollback segment to twice the total data volume of the account batch. You can estimate the account batch data volume by multiplying the average number of events per batch with the batch size.

> **Note**
>
> Use the EVENT_T and major child tables, such as EVENT_BAL_IMPACTS_T, to determine the average number of events per batch. Typically, 90% of the account batch data volume can be attributed to event data.

- Set the optimal size for the rollback segments to twice the initial size.
- Set the next size for the rollback segments to half the initial size.
- Set the maximum number of extends to unlimited.

For more information, see your Oracle documentation. For information on additional ways to tune your database for AMM, contact your Oracle BRM representative.

5

Using Account Migration Manager

Learn how to use the Oracle Communications Billing and Revenue Management (BRM) Account Migration Manager (AMM) software to migrate accounts from a source database schema to a destination database schema in the same database.

Topics in this document:

Overview of Account Migration Tasks

Migrating accounts includes the following general tasks. Although you can perform some tasks at any time, the following order is recommended:

1. Create the account search configuration file.

   See "Creating the Account Search Configuration File" for more information.

2. Submit the account migration job.

   See "Submitting the Account Search File" for more information.

3. For account group migration, run a **group_details** report to verify that each account group includes all account members.

   See "Checking Account Group Details" for more information.

   > **Note**
   >
   > You must verify that the job includes all accounts in the account group. Any missing accounts will be stored in a separate database schema from the account group, which severs the account's relationship with the group.

4. Enable the account migration job.

   See "Enabling Migration Jobs in the Queue" for more information.

5. Start the AMM Controller.

   See "Starting the AMM Controller" and "Monitoring the AMM Controller" for more information.

6. Monitor the job's progress.

   See "Monitoring Account Migration" for more information.

7. Fix account migration failures, when necessary.

   See "Handling Account Migration Failures" for more information.

8. Purge the migrated accounts from the source database schema.

   See "Purging Migrated Objects from the Source Database Schema" for more information.

9. Stop the AMM Controller.

   See "Stopping the AMM Controller" for more information.

See "Deleting Jobs from the Source Database Schema" for information on deleting jobs from the source database schema.

Creating the Account Search Configuration File

You use the account search configuration file to specify the source and destination database schemas, the search criteria, the maximum number of accounts in a job, and the number of accounts in each batch.

AMM can search for accounts that meet five default criteria:

- Account creation date
- Account status
- Billing day of month
- Charge offer name
- POID

If you would like to migrate accounts that meet custom criteria, see "Creating Custom Account Search Criteria".

To create an account search configuration file:

1. Copy the sample account search configuration file (*BRM_home*/**apps/amt/account_search.cfg**) and save it with another name. Use this file, which contains all of the configuration entries, as a template.
2. Edit the entries listed in Table 5-1 in the file.

> **Note**
>
> Only the source database schema, destination database schema, batch size, and one other entry is required. If you do not want to use an entry, leave it blank.

Table 5-1 account_search.cfg Parameters

Parameter	Description	Required
src_database	Specifies the source database schema, which is the schema from which you are migrating accounts. For example, enter **0.0.0.1**. This value must match one of the database numbers specified in the **Infranet.properties** file.	YES
dest_database	Specifies the destination database schema, which is the schema to which you are migrating accounts. For example, enter **0.0.0.2**. This value must match one of the database numbers specified in the **Infranet.properties** file.	YES
start_creation_date	Use this parameter to migrate accounts that were created in a specific date range. AMM migrates accounts created between midnight (00:00:00) on the start date and 23:59:59 on the end date. For example, to migrate accounts created after midnight on August 1, 2004, enter **08/01/2004**. **Important:** If you set this parameter, you must also set the **end_creation_date** parameter.	no
end_creation_date	Use this parameter to migrate accounts that were created in a specific date range. AMM migrates accounts created between midnight (00:00:00) on the start date and 23:59:59 on the end date. For example, to migrate accounts created on or before 11:59:59 p.m. on August 10, 2004, enter **08/10/2004**. **Important:** If you set this parameter, you must also set the **start_creation_date** parameter.	no
migration_mode	Specifies whether to migrate account groups. When AMM finds an account that belongs to a hierarchical account or charge or discount sharing group, AMM migrates all accounts related to that account. • **IncludeAccountGroup** specifies to migrate account groups. • **ExcludeAccountGroup** specifies to exclude account groups from migrations. The default is **ExcludeAccountGroup**. **Important:** If you set this parameter, you must also set the **max_group_size** parameter.	no
max_group_size	Specifies the maximum size of an account group that AMM can migrate. If an account group exceeds the maximum number of accounts, AMM excludes the account group from the job. The default is **100**.	no
product_name	Migrates accounts that purchased the specified charge offer. For example, **Offer 1b - Email Account**.	no
account_status	Migrates accounts based on the specified account status: • **Active** specifies to migrate active accounts only. • **Inactive** specifies to migrate inactive accounts only. • **Closed** specifies to migrate closed accounts only.	no
bill_day_of_month	Migrates accounts that have the specified billing day of month. You can specify any number from 1 through 31. For example, enter **4** to migrate all accounts that are billed on the 4th of the month.	no
max_accounts	Specifies the maximum number of accounts to move in a job.	no

Table 5-1 (Cont.) account_search.cfg Parameters

Parameter	Description	Required
batch_size	Specifies the number of accounts in each batch. You can specify any amount from 1 through 1,000. However, for optimal performance, set this to an integer between 50 and 100. **Important:** • Using a batch size of more than 50 accounts does not improve performance. • If you set this to a number greater than 100, you must increase the size of your Oracle rollback segments. For more information, contact your Oracle BRM representative.	YES
poid_list	Migrates accounts based on the POID. Use comma separators, for example, **22860, 22861, 22862**. Limit the number of accounts to 1,000 or less.	no
cross_schema_group	Specifies whether to migrate accounts that are members of a sharing group: • **Enabled**: AMM checks whether an account member is a member of a cross-schema sharing group. If an account is a member, AMM does not migrate it. • **Disabled**: AMM does not check whether an account is a member of a cross-schema sharing group. The default is **Disabled**.	no

3. Save the file.

Sample Account Search Configuration File

The following sample account search configuration file specifies to:

- Migrate accounts from database schema **0.0.0.1** to database schema **0.0.0.2**.
- Migrate in batches of 50 accounts.
- Migrate only nonmember accounts.
- Migrate accounts that meet the following criteria:
  - Created between January 1, 2004 and June 30, 2004
  - Have an active account status
  - Purchased the **Offer 1b - Email Account** charge offer

```
src_database=0.0.0.1
dest_database=0.0.0.2
start_creation_date=01/01/2004
end_creation_date=06/30/2004
migration_mode=ExcludeAccountGroup
max_group_size=
product_name=Offer 1b - Email Account
account_status=Active
bill_day_of_month=
max_accounts=
batch_size=50
poid_list=
```

Submitting the Account Search File

When you submit an account search file, the **pin_amt** utility searches the source database schema and populates the job management tables on the primary, source, and destination database schemas with a list of accounts meeting the specified criteria.

1. Submit your account search information to the **pin_amt** utility:

   ```
   % pin_amt -s AccountSearchFile
   submitted job
   job_id=30
   ```

   The **pin_amt** utility notifies you if it successfully submitted the file and gives you the job ID number.

2. Write down the job ID number, because you will need it later.

Enabling Migration Jobs in the Queue

The AMM Controller can only begin processing an account migration job after it's enabled in the queue.

To enable a job in the queue, enter this command:

```
% pin_amt -e JobID
enabled job
```

Starting the AMM Controller

After it is started, the AMM Controller runs as a server process, continuously checking for jobs to process in the queue.

To start the AMM Controller:

1. Go to the *BRM_home***/sys/amt** directory.

   > **Note**
   >
   > The default location of the AMM **Infranet.properties** file is *BRM_home***/sys/amt**. If the AMM **Infranet.properties** file is located in a different directory, start the AMM controller from that location.

2. Run the following command:

   ```
   % pin_amt -c start [-a ControllerID]
   controller is started
   controller_id=1
   ```

> **Note**
>
> If your system contains multiple AMM Controllers, use the **-a** option to specify which AMM Controller to start. By default, **pin_amt** starts Controller 1.

The **pin_amt** utility notifies you if the AMM Controller started successfully and which AMM Controller is active.

Monitoring the AMM Controller

You can monitor the AMM Controller's status at any time by using the **pin_amt** utility or checking the AMM Controller log file.

Checking the AMM Controller Status

To check whether the AMM Controller is up and running, enter this command:

```
% pin_amt -c status [-a ControllerID]
controller status is up
```

> **Note**
>
> If your system contains multiple AMM Controllers, use the **-a** option to specify which AMM Controller to check.

The **pin_amt** utility notifies you that the AMM Controller is up or down. If the AMM Controller is down or cannot be started, check the AMM Controller log file for more information.

Checking the AMM Controller Log File

The AMM Controller log file contains a detailed list of all transactions processed by the AMM Controller. This log is created in the directory specified in the **controller_*N*_log_directory** entry of the **Infranet.properties** file. You can open the log file by using a text editor.

Monitoring the AMM Controller in Real Time

You can use the **pin_amt** utility to see what the AMM Controller is doing in real time.

To monitor the AMM Controller in real time, enter this command:

```
% pin_amt -c log [-a ControllerID]
```

> **Note**
>
> If your system contains multiple AMM Controllers, use the **-a** option to specify which AMM Controller to check.

A separate Xterm window opens. For best viewing, set the Xterm width to 120. If an Xterm window fails to open, make sure your **DISPLAY** environment variable is set correctly.

Monitoring Account Migration

You can monitor the status of jobs in the queue by running three special AMM reports: **list_jobs**, **job_details**, and **group_details**.

Monitoring Job Status

The **list_jobs** report provides the status of each job in the queue, including the number of batches that failed to migrate.

To run the **list_jobs** report, enter this command:

```
% pin_amt -r list_jobs
```

Sample output from a **list_jobs** report:

```
Tue Mar 12                                                                                         page 1
                                                AMT jobs

                               Total   Failed    Succ.
                             Account  Account  Account                  Proc.  Job creation
Job Name  User Name  Job ID  batches  batches  batches  Job Status  time[sec]  time              Accounts
--------  ---------  ------  -------  -------  -------  ----------  ---------  ----------------  --------
test.cfg  pin             1        6        0        6  FINISHED           40  02/02/2002 13:39       100
srch.cfg  pin             2        3        0        3  FINISHED           25  02/15/2002 12:00        50
mar1.cfg  pin             3        8        1        7  FINISHED          205  03/01/2002 18:42       400
```

If any batches failed, you can see greater detail on why the batch failed by running the **job_details** report.

Checking Job Details

The **job_details** report provides detailed information about a job's status, including why a batch failed.

To run the **job_details** report, enter this command:

```
% pin_amt -r job_details
enter job id:
```

Sample output from a **job_details** report:

```
Tue Mar 12                                                                                         page 1
                                             AMT job details

        Account                                         Processing start  Batch processing
Job ID  batches  Status    Error Message                date                     time[sec]  Accounts
------  -------  --------  ---------------------------  ----------------  ----------------  --------
     3        1  FINISHED                               03/01/2002 18:42                25        50
              2  FAILED    ORA-02055: distributed       03/01/2002 18:42                 5         0
                           update operation failed;
                           rollback required
                           ORA-02049: timeout:
                           distributed transaction
                           waiting for lock
                           ORA-06512: at “PIN.AMT_MV",
                           line 454
                           ORA-06512: at line 1
              3  FINISHED                               03/01/2002 18:42                25        50
              4  FINISHED                               03/01/2002 18:43                25        50
              5  FINISHED                               03/01/2002 18:43                25        50
              6  FINISHED                               03/01/2002 18:44                25        50
              7  FINISHED                               03/01/2002 18:44                25        50
              8  FINISHED                               03/01/2002 18:45                25        50
```

The report lists any error messages from the Oracle database. For information, see the Oracle documentation.

Checking Account Group Details

The **group_details** report lists the accounts in each account group and provides information about each group's migration status. You use this information to verify that all account members are included in a group.

Note

All accounts in a hierarchical account or charge or discount sharing group must reside in the same database schema. Any accounts separated from a parent account will no longer be associated with the account group.

To run the **group_details** report, enter this command:

```
% pin_amt -r group_details
enter job id:
enter group id:
```

Sample output from a **group_details** report:

```
Tue Mar 12                                                                                         page 1
                                           AMT group details

        Account  Batch     Group      Group
Job ID  batches  Status    ID         Status
------  -------  --------  ---------  ----------------
     3        1  FINISHED  1          NOT_PROCESSED


Tue Mar 12                                                                                         page 1
                                       AMT group member details

Account
batches  Accounts ID  Accounts DB
-------  -----------  -----------
      1  17009        2
      1  17289        2
      1  16489        2
      1  17313        2
      1  16465        2
      1  17066        2
```

Handling Account Migration Failures

An account batch may fail for several reasons. The most common reasons are as follows:

- An application is accessing or modifying the data you are attempting to migrate.
- The database is down.

Finding Debugging Information

For information on why a batch failed, you can run a **job_details** report or check any of the following files, which are located in the directories you specified in the **Infranet.properties** file.

- AMM installation log file (**pin_amt_install.log**)
- AMM Controller log file (**controller**_*N_YYYYMMDDhhmm*.**log**)
- **pin_amt** log file (**pin_amt.log**)
- AMM configuration file (**Infranet.properties**)
- Account search configuration file (**account_search.cfg**)
- AMM Mover log files (**amt_migrate**_*JobID_BatchNumber*.**log**)

- AMM delete log file (**amt_delete_***JobID*_*BatchNumber***.log**)

If you need assistance in resolving migration failures, send these files along with any additional information about the problem to your Oracle BRM representative.

Reprocessing Failed Batches

To reprocess a batch that failed:

1. Fix the problem.
2. Change the status of the batch from FAILED to NOT PROCESSED:

   ```
   % pin_amt -b JobID:BatchNumber
   ```

3. Enable the job in the queue again:

   ```
   % pin_amt -e JobID
   ```

   The AMM Controller processes all batches that have a NOT PROCESSED status and ignores batches with a FINISHED status.

Purging Migrated Objects from the Source Database Schema

After you successfully migrate your accounts, you can improve your overall system performance by purging the migrated (invalid) objects from your source database schema. Also, because the purging process uses only one thread, purges accounts sequentially, and doesn't affect data used by BRM, you can purge accounts at any time.

To purge successfully migrated objects from the source database schema, enter this command:

```
pin_amt -p SourceDatabaseSchema
```

Deleting Jobs from the Source Database Schema

You can use the delete option to:

- Remove both failed and successfully migrated jobs from your database schemas
- Free up disk space

The delete option performs the following actions listed in Table 5-2:

Table 5-2 Delete Job Actions

Job Type	Action
Failed jobs	Deletes the job from the AMM job management tables.
Successfully migrated jobs	• Deletes the job from the AMM job management tables. • Deletes account-related data from the source database schema.

To delete a job, run the **pin_amt** script with the delete option:

```
pin_amt -d JobID
```

Stopping the AMM Controller

You can stop the AMM Controller at any time. If you stop the AMM Controller while it is processing a batch, it finishes the batch before stopping.

To stop the AMM Controller, enter this command:

```
% pin_amt -c stop
controller is stopped
controller_id=1
```

Pausing and Resuming Account Migration

If a job contains a large number of accounts, but you only have a limited amount of time in which to migrate accounts, you can migrate the job in stages. The AMM software allows you to start an account migration job and then pause it when your window of opportunity is over. When you reach the next window of opportunity, you can resume the job where it left off.

> **Note**
>
> An AMT Controller must completely finish migrating one job before it can start migrating another job. Therefore, if you pause one job and then enable a second job, the AMT Controller cannot begin processing the second job until the first job is finished.

To pause an account migration job, enter this command:

```
% pin_amt -c pause
paused controller
controller_id=1
```

To resume an account migration job, enter this command:

```
% pin_amt -c continue
continued controller
controller_id=1
```

Automating Account Migration

You can use an external scheduler, such as **cron**, to automate account migration during your maintenance window. If you need to migrate a large number of accounts, you can set up **cron** to stop and restart account migration at specific times.

For example, scheduling account migration for every Sunday from 2:00 a.m. to 4:00 a.m. requires these tasks:

1. Stop the AMM Controller.
2. Create your account search configuration files.
3. Submit your jobs.
4. Enable your jobs in the queue.
5. Configure one **cron** job to start the AMM Controller every Sunday at 2:00 a.m. and check for errors. See "AMM Return Codes and Messages".

6. Configure a second **cron** job to stop the AMM Controller every Sunday at 4:00 a.m. and check for errors.

6

Modifying Applications to Work with AMM

Learn how to modify your custom client applications and custom reports to work with the Oracle Communications Billing and Revenue Management (BRM) Account Migration Manager (AMM) software.

Topics in this document:

Enabling ECE to Rate Events during Account Migration

When you enable ECE to rate events during account migration, the AMM business events listed in Table 6-1 automatically notify ECE that an account migration job is occurring. They enable ECE to:

- Track the status of the migration
- Notify BRM if ECE fails to clear its rated event data store before the migration begins (a migration prerequisite)
- Continue rating events while the accounts are migrated
- Update its information about the target database schema for successfully migrated accounts

Table 6-1 ECE Response to AMM Business Events

AMM Business Event	ECE Response
HoldCDRProcessing	1. Gets the migration job ID, the source database schema, and the target database schema from this event. 2. Queries the BRM database for the list of accounts that belong to the migration job. 3. Waits for all existing rated events associated with those accounts to be extracted from the Oracle NoSQL database data store. 4. Does one of the following: If the extraction succeeds: – Assigns the IN_ACCOUNT_MIGRATION status to the accounts. – Updates their target database schema information. – Sends an ACKHoldCDRProcessing acknowledgment to the BRM acknowledgment queue. – Continues rating incoming usage events for the migrated accounts but does not extract them from the Oracle NoSQL database data store. If the extraction fails, ECE sends a NACKHoldCDRProcessing acknowledgment to BRM, and BRM does not migrate the accounts.

Table 6-1 (Cont.) ECE Response to AMM Business Events

AMM Business Event	ECE Response
MigrateAcct	Sends an ACKMigrateAcct acknowledgment to the AMM acknowledgment queue.
MigrateSource	Sends an ACKMigrateSource acknowledgment to the AMM acknowledgment queue.
MigrateDestination	Sends an ACKMigrateDestination acknowledgment to the AMM acknowledgment queue.
ResumeCDRProcessing	1. Gets the migration job ID, the source database schema, and the target database schema from this event. 2. Queries the BRM database for the list of accounts that belong to the migration job. 3. Removes the IN_ACCOUNT_MIGRATION status from those accounts. 4. Loads all the rated events that were generated while the accounts' status was IN_ACCOUNT_MIGRATION into the new target database schema.

To enable ECE to rate events during account migration:

1. Configure the definition of your system's AMM controllers for ECE:
   a. Go to the *BRM_home***/sys/amt** directory and open the **Infranet.properties** file in a text editor.
   b. Verify that the following entry is set to **true**:

      ```
      controller_1_event_generation=true
      ```

   c. If more than one controller is defined in the file, ensure that each controller's **controller_***controller_number***_event_generation=** entry is set to **true**.
   d. Save and close the file.

      The change takes effect the next time the **pin_amt** utility is run. For more information, see the discussion about that utility in *BRM System Administrator's Guide*.
2. Configure ECE to use the AMM acknowledgment queue.
3. Verify that Customer Updater is correctly configured for your system and running.

   ECE receives AMM business events from the BRM Account Synchronization DM database queue through Customer Updater. Because account migration involves multiple database schemas, Customer Updater must be configured to support all the database schemas in your system. It must also be configured to send AMM-related acknowledgments from ECE to your system's AMM acknowledgment queue.
4. Verify that BRM Gateway is correctly configured for your system and running.
5. Verify that Rated Event Formatter is correctly configured for your system and running.

Configuring ECE to Use the AMM Acknowledgment Queue

The AMM acknowledgment queue is used to send AMM-related acknowledgments from ECE to BRM; it is an Oracle AQ database queue on the BRM system.

To configure ECE to use the AMM acknowledgment queue:

1. Access the ECE configuration MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).

   c. Connect to the ECE charging server node enabled for JMX management.

      This is the charging server node set to **start CohMgt = true** in the *ECE_home***/oceceserver/config/eceTopology.conf** file.

   d. Start a JMX editor that enables you to edit MBean attributes, such as JConsole.

   e. In the editor's MBean hierarchy, find the ECE configuration MBeans.

2. Expand the **ECE Configuration** node.
3. Expand **charging.connectionConfigurations.oracleQueueConnection**.
4. Expand **Attributes**.
5. Set the **amtAckQueueName** attribute to the fully qualified name of the acknowledgment queue to which the **pin_amt** utility listens to AMM-related acknowledgment events:

   ```
   schema.ammAcknowledgmentQueue
   ```

   where:

   - *schema* is the name of the BRM database schema in which the AMM acknowledgment queue resides.
   - *ammAcknowledgmentQueue* is the name of the AMM acknowledgment queue.

   For example:

   ```
   PIN01.ECE_AMT_ACK_QUEUE
   ```

6. Save your changes.

Modifying Custom Client Applications for AMM

Custom client applications that connect to a specific database schema and try to access an object based on a POID may receive a PIN_ERR_INVALID_OBJ error if the object was migrated to another database schema. You must modify any custom client applications to handle that error and then perform a global search to find the object's correct location.

To obtain the correct POID of an object, modify your application to call the PCM_OP_GLOBAL_SEARCH opcode from its exception handling routine.

This example shows a call to the PCM_OP_GLOBAL_SEARCH opcode when the PIN_ERR_INVALID_OBJ error is returned from the Oracle DM:

```
/* Error? */

if (PIN_ERR_IS_ERR(ebufp)) {
PIN_ERR_LOG_EBUF(PIN_ERR_LEVEL_ERROR,
"sample_read_obj_search error", ebufp);
}
/* Call the DM to do a global search.*/

PCM_OP(ctxp, PCM_OP_GLOBAL_SEARCH, 0, flistp, &r_flistp, ebufp);

return;
```

The following opcodes return the PIN_ERR_INVALID_OBJ error when a POID specified in an input flist is invalid:

- PCM_OP_READ_OBJ
- PCM_OP_READ_FLDS

- PCM_OP_WRITE_FLDS
- PCM_OP_INC_FLDS
- PCM_OP_DELETE_OBJ
- PCM_OP_DELETE_FLDS
- PCM_OP_TRANS_OPEN

Modifying Custom BRM Reports for AMM

After account migration, any custom BRM reports created prior to BRM Release 6.2 ServicePak1 might retrieve and process duplicate data from your source and destination database schemas. For example, if an account object is migrated from database schema **0.0.0.1** to database schema **0.0.0.2**, your report might retrieve the account object from both database schemas.

To prevent this, use the Oracle Business Intelligence Publisher to add the following line to the WHERE clause of each custom report's query:

```
TABLE_T.POID_DB  >  0
```

where *TABLE_T* satisfies these conditions:

- It is a database table used by the report.
- It is one of the tables moved from the source database schema to the destination database schema when account data is migrated.
- It is associated with every record the report must retrieve.

> **Note**
>
> If a single table does not satisfy the last condition, add the same line for several tables that together satisfy the last condition.

AMM Return Codes and Messages

AMM uses the return codes and messages shown in Table 6-2. To automate account migration, you can modify your external application to check for the following return codes and respond appropriately.

Table 6-2   AMM Return Codes and Messages

Return Code Number	Return Code	Return Message
100	CONTROLLER_STARTED_SUCC	controller is started
101	CONTROLLER_STOPPED_SUCC	controller is stopped
102	CONTROLLER_PAUSED_SUCC	paused controller
103	CONTROLLER_CONTINUED_SUCC	continued controller
104	CONTROLLER_UP_SUCC	controller status is up
105	CONTROLLER_DOWN_SUCC	controller status is down

Table 6-2 (Cont.) AMM Return Codes and Messages

Return Code Number	Return Code	Return Message
106	SUBMIT_JOB_SUCC	submitted job
107	DELETE_JOB_SUCC	deleted job
108	PURGE_DATABASE_SUCC	purged database
109	ENABLE_JOB_SUCC	enabled job
110	REPORT_SUCC	generated report
111	ENABLE_BATCH_SUCC	enabled batch
200	CONTROLLER_RUNNING_ERROR	ERROR: controller is already running
201	CONTROLLER_PAUSED_ERROR	ERROR: controller is already paused
202	CONTROLLER_SPEC_ACCESS_ERROR	ERROR: controller specification does not exist
203	CONTROLLER_COMM_ERROR	ERROR: controller cannot be reached
204	SEARCH_SPEC_IO_ERROR	ERROR: account search specification could not be accessed
205	OPERATION_ROLLBACK_ERROR	ERROR: operation rollback
206	SEARCH_SPEC_PARSE_ERROR	ERROR: account search specification cannot be parsed
207	OPERATION_PERM_ERROR	ERROR: operation not permitted for current user OR job_id/batch_id does not exist
208	CONTROLLER_UNKNOWN_HOST_ERROR	ERROR: controller host not found
209	CONTROLLER_PROCESS_ERROR	ERROR: controller process could not be created
210	REPORT_PROCESS_ERROR	ERROR: external process interruption
211	REPORT_SCRIPT_ACCESS_ERROR	ERROR: reporting tool not found or report type does not exist
212	OPT_PARAM_REQ_ERROR	ERROR: one optional parameter is required
213	CONFIG_FILE_ACCESS_ERROR	ERROR: configuration file cannot be accessed
214	INIT_ERROR	ERROR: could not create new object
215	EMPTY_RESULTSET_ERROR	ERROR: account search resulted in 0 accounts, job submission failed
216	CONVERSION_CLASS_LOAD_ERROR	ERROR: dynamic loading of custom Conversion class failed

7

Modifying the Account Migration Manager

Learn how to create custom search criteria for the Oracle Communications Billing and Revenue Management (BRM) Account Migration Manager (AMM).

Topics in this document:

- Creating Custom Account Search Criteria

Creating Custom Account Search Criteria

AMM allows you to migrate accounts that meet custom criteria. For example, you can create custom criteria for finding and migrating accounts located in a certain state or belonging to a particular service provider.

To create a custom search criteria, perform these tasks:

1. Creating a Search Template
2. Adding New Entries to the Account Search Configuration File
3. Implementing and Compiling the Conversion Interface
4. Verifying Your Search Criteria

Creating a Search Template

AMM searches for accounts in a database schema by using SQL statements generated from an account search template. Before AMM can generate a SQL statement with new search criteria, you must first create a template for it in the custom account search properties file.

To create a template for your search criteria:

1. Open the custom account search properties file (*BRM_home***/apps/amt/com/portal/amt/custom_account_search.properties**) in a text editor.
2. Add SQL fragments for your search criteria by using the following syntax:

   ```
   criteria_name=AND SQL_condition \n
   ```

   Where:

   - *criteria_name* is the name of your selection criteria.
   - *SQL_condition* is a valid SQL *condition* that searches a BRM table and references one or more search variables, as shown below. Search variables must be surrounded by curly braces "**{ }**" and match an entry in the **account_search.cfg** file.

     ```
     condition_text '{SearchVariable}'...
     ```

> **Note**
>
> *SearchVariable* must use a unique name and must not match one of the BRM-defined search variable names. For the list of BRM-defined search variables, see "Creating the Account Search Configuration File".

For information on the SQL condition, see your Oracle documentation.

3. Save and exit the file.

Sample Search Template

The following sample search template enables AMM to search for accounts located in a particular state. It tells AMM to search the ACCOUNT_NAME_INFO_T table for objects with the **state** field set to a specified value.

```
# select accounts based on state
cust_acct_search_account_state_constraint=\
AND EXISTS \n\
(SELECT an.obj_id0 FROM account_nameinfo_t an \n\
WHERE an.obj_id0 = a.poid_id0 and an.state = '{account_state}') \n
```

Adding New Entries to the Account Search Configuration File

When building a query, AMM replaces the search variables in your account search template with values from the account search configuration file (*BRM_home***/apps/amt/account_search.cfg**).

To add an entry for your search variable:

1. Open the *BRM_home***/apps/amt/account_search.cfg** file in a text editor.
2. Add your new search entry and comments to the file.

   > **Note**
   >
   > *SearchVariable* must match the search variable name referenced in the **custom_account_search.properties** file.

   ```
   # - You should add comments about the new search entry and
   #   valid values.

   SearchVariable=
   ```

3. Save and exit the file.

Sample Account Search Configuration File

A sample search entry for the **account_state** search criteria:

```
# - Migrates accounts located in a specific state. Valid values
# are California and Oregon.
account_state=
```

Implementing and Compiling the Conversion Interface

Each custom search variable must have a corresponding Java implementation of the **Conversion** interface.

1. Run the appropriate profile script for your shell. This script sets your CLASSPATH and PATH environment variables to the appropriate values. For example, for the c shell:

   ```
   % cd BRM_home/apps/amt
   % source profile.csh
   ```

2. Create a class that implements the **Conversion** interface.
3. Save and compile your *SearchVariable***.java** source file in the *BRM_home***/apps/amt/com/portal/amt** directory.

   ```
   % cd BRM_home/apps/amt/com/portal/amt
   % javac SearchVariable.java
   ```

   This creates a *SearchVariable***.class** file in the same directory.

> **Note**
>
> For AMM to successfully build a search with your custom search criteria:
>
> - The class name must match the search variable name used in the **custom_account_search.properties** and **account_search.cfg** files.
> - The class must reside in the *BRM_home***/apps/amt/com/portal/amt** directory.

Sample Class Implementing Conversion Interface

The following sample class, **account_state.class**, allows users to search for accounts from California or Oregon.

```
package com.portal.amt;
public class account_state implements Conversion {
  public String convert(String stateName) throws ConversionException {
    String stateCode = null;
    if(stateName.equals("California")) {
      stateCode = "CA";
    } else if(stateName.equals("Oregon")) {
      stateCode = "OR";
    } else {
      throw new
        ConversionException("Error: account_state " + stateName + " unknown.");
    }
    return(stateCode);
  }
}
```

Verifying Your Search Criteria

Before migrating accounts with the new search criteria, verify its accuracy by:

1. Verifying That the Search Criteria Creates Valid SQL Statements
2. Verifying That the Search Criteria Finds Correct Accounts

Verifying That the Search Criteria Creates Valid SQL Statements

Use the "pin_amt_test" utility to verify that your custom search template generates a valid SQL statement.

1. Open your account search configuration file (*BRM_home***/apps/amt/account_search.cfg**) in a text editor.
2. Enter values for the source and destination database schemas, the batch size, and your custom search criteria.

   For example, you might enter the following to test the **account_state** criteria:

   ```
   src_database=0.0.0.1
   dest_database=0.0.0.2
   start_creation_date=
   end_creation_date=
   product_name=
   account_status=
   bill_day_of_month=
   max_accounts=
   batch_size=50
   poid_list=
   account_state=California
   ```

3. Save and exit the file.
4. Use the **pin_amt_test** utility to generate a SQL statement with your new search criteria.

   ```
   % pin_amt_test -c AccountSearchFile
   ```

   If successful, the utility displays the resulting SQL statement. For example, the sample **account_state** criteria generates the following:

   ```
   Compile: account_search.cfg
   ------
   account search SELECT statement:
   -- acct_search_select: default
   SELECT
   DISTINCT a.poid_id0
   FROM account_t a
   WHERE
   ...
   ```

   If the compilation failed, the utility returns the file name and line number where the error occurred. For example, the utility returns the following when users enter an invalid state for the sample **account_state** criteria:

   ```
   Compile: account_search.cfg
   --------
   account_search.cfg:32: mapping of account_state field value Florida failed
   ```

5. Verify that the resulting SQL statement is correct.

Verifying That the Search Criteria Finds Correct Accounts

Use the **pin_amt_test** utility to verify that your search criteria works properly. This utility only displays results on the screen and does not migrate your objects.

To verify your search query:

1. Create a database schema with a precisely defined set of account data. The database schema should contain a small number of accounts.
2. Create a list of account POIDs that meet the custom criteria you are testing. For example, write down the POIDs of all accounts created in California.
3. Open your account search configuration file (*BRM_home*/**apps/amt/account_search.cfg**) in a text editor.
4. Enter values for the source and destination database schemas, the batch size, and your custom search criteria.

   For example, you might enter the following to test the **account_state** criteria:

   ```
   src_database=0.0.0.1
   dest_database=0.0.0.2
   start_creation_date=
   end_creation_date=
   product_name=
   account_status=
   bill_day_of_month=
   max_accounts=
   batch_size=50
   poid_list=
   account_state=California
   ```
5. Save and exit the file.
6. Use the **pin_amt_test** utility to run your search query against the source database schema:

   ```
   % pin_amt_test -e AccountSearchFile
   ```
7. The utility prints to the screen a list of account POIDs that meet your search criteria. Compare this list of POIDs with the list you created in step 2.

   If the lists match, your new search criteria works properly and you can start using it to migrate accounts.

   If the lists do not match, make sure:

   - Your search template generates a valid SQL statement.
   - Your search template, search configuration file, and class all refer to the same variable name.

8 Account Migration Utilities

Learn about the syntax and parameters for the Oracle Communications Billing and Revenue Management (BRM) account migration utilities.

Topics in this document:

- pin_amt
- pin_amt_test

pin_amt

Use this utility to migrate accounts from a source database schema to a destination database schema in the same BRM database.

You define which accounts to migrate in the account search configuration file in the *BRM_home***/apps/amt** directory. See "Creating the Account Search Configuration File".

> **Note**
>
> To connect to the BRM database, the **pin_amt** utility needs a configuration file in the *BRM_home***/sys/amt** directory. For information on how to create this file, see "Connecting AMM to Your Database Schemas".

Location

*BRM_home***/bin**

Syntax

```
pin_amt    [-f ConfigFileName] [-a ControllerID]
           [-c start | stop | pause | continue | status | log]
           [-s AccountSearchFile] [-d JobID]
           [-r list_jobs | job_details | group_details]
           [-p SourceDatabaseSchema] [-e JobID][-b JobID:BatchNumber][-h]
```

Parameters

-f *ConfigFileName*
Specifies the name of the configuration file that defines how to connect to each database schema in your system. By default, the **pin_amt** utility looks in the *BRM_home***/sys/amt** directory. If your configuration file is located in a different directory, you must specify the entire path for the file.
If you use the *BRM_home***/sys/amt/Infranet.properties** file, you can ignore this parameter.

-a *ControllerID*
Specifies the AMM Controller to use.
Use this option only with the **-c** option and only when your system contains multiple AMM Controllers. If your system contains only one AMM Controller, ignore this option.

-c start | stop | pause | continue | status | log
Sets the AMM Controller. When your system contains multiple AMM Controllers, you must also use the **-a** option.
Use one of these options with the parameter:
-c start starts the AMM Controller.
-c stop stops the AMM Controller. If you stop the AMM Controller while it is processing a batch, it finishes processing the batch before stopping.
-c pause pauses the processing of a job in the queue. If you pause the Controller while it is processing a batch, it finishes processing the batch before pausing.
-c continue restarts processing a job that was paused.
-c status displays the current status of the AMM Controller.
-c log displays all AMM Controller transactions in real time through an Xterm window. To use this option, you must set the **DISPLAY** environment variable correctly.

-s *AccountSearchFile*
Specifies the name of the configuration file that defines which accounts to migrate. For information on how to create the file, look at the sample account search configuration file (*BRM_home***/apps/amt/account_search.cfg**).
By default, the **pin_amt** utility looks in the current working directory. If your configuration file is located in a different directory, you must specify the entire path for the file.

-d *JobID*
Deletes the specified job from the job management tables. When deleting a job that migrated successfully, this option also purges all migrated accounts from the source database schema.

-e *JobID*
Enables the specified job in the queue.

-r list_jobs | job_details | group_details
Runs the preconfigured report. Use one of these options with the parameter:
-r list_jobs displays the status of all jobs currently in the queue.
-r job_details displays the details of the specified job.
-r group_details displays the details of the specified account group.

-p *SourceDatabaseSchema*
Purges all accounts that were successfully migrated from the source database schema. For example, to purge invalid objects from your primary schema, enter the following:

```
pin_amt -p 0.0.0.1
```

-h
Displays the syntax and parameters for this utility.

-b *JobID*:*BatchNumber*
Changes the status of the batch from FAILED to NOT PROCESSED, and the job from FINISHED to DISABLED. For information, see "About Batch Status Flags" and "About Job Status Flags".

Results

The **pin_amt** utility notifies you when it successfully completes a command.

For error information about each job, run a report or look in the AMM Mover log file. The log file is in the directory specified by the **0.0.0.*x*_mover_log_file_dir** entry in the **Infranet.properties** file.

For error information about the AMM Controller, look in the AMM Controller log file. The log file is in the directory specified by the **controller_*N*_log_directory** entry in the **Infranet.properties** file.

The history of all **pin_amt** commands is located in the **pin_amt** log file.

pin_amt_test

Use this utility to test your custom BRM account search criteria. This utility safely runs your search criteria against a source database schema and displays either a SQL SELECT statement or a list of account POIDs meeting your search criteria.

You define which custom search criteria to test in the account search configuration file (*BRM_home***/apps/amt/account_search.cfg**). See "Creating the Account Search Configuration File".

> **Note**
>
> To connect to the BRM database, the **pin_amt_test** utility needs a configuration file in the *BRM_home***/sys/amt** directory. For information on how to create this file, see "Connecting AMM to Your Database Schemas".

Location

*BRM_home***/apps/amt**

Syntax

```
pin_amt_test [-f ConfigFileName ]
             -c AccountSearchFile | -e AccountSearchFile | -h
```

Parameters

-f *ConfigFileName*
Specifies the name of the configuration file that defines how to connect to each database schema in your system. By default, the **pin_amt_test** utility looks in the *BRM_home***/sys/amt** directory. If your configuration file is located in a different directory, you must specify the entire path for the file.
If you use the *BRM_home***/sys/amt/Infranet.properties** file, you can ignore this parameter.

-c *AccountSearchFile*
Displays the SQL SELECT statement generated with the account search criteria specified in *AccountSearchFile*.
By default, the **pin_amt_test** utility looks in the current working directory. If your account search file is located in a different directory, you must specify the entire path for the file.

-e *AccountSearchFile*
Runs the SQL SELECT statement for the specified search criteria against the source database schema and displays the list of account POIDs meeting the search criteria.
By default, the **pin_amt_test** utility looks in the current working directory. If your account search file is located in a different directory, you must specify the entire path for the file.

-h
Displays the syntax and parameters for this utility.

Results

The **pin_amt_test** utility prints to the screen the SQL SELECT statement, a list of accounts meeting your search criteria, or an Oracle error message. For information about Oracle error messages, see your Oracle documentation.

A

AMM Job Management Tables

This appendix describes the tables used for account migration in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About AMM Job Management Tables

About AMM Job Management Tables

Job management tables are created on your database schemas during installation and are populated with information about each migration job.

The AMM installer creates the tables listed in Table A-1 on your schemas:

Table A-1   AMM Job Management Tables

Table Name	Schema	Description	When Populated
AMT_ACCOUNT_BATCH_TABLE_T	Primary only	Stores the list of tables containing data to migrate for a particular batch.	Populated by the AMM Mover when it migrates a particular batch.
AMT_METADATA_T	All schemas	AMM data dictionary. This lists all default BRM tables as well as any custom tables you created. If you add any tables after you install AMM, you must refresh the AMM data dictionary. See "Configuring AMM for New Custom Tables".	During installation and when you run **pin_amt_install.pl -m** to refresh the AMM data dictionary.
AMT_POID_TYPE_MAP_T	All schemas	Maps the POID type to the table name. This table is static.	During installation and when you run **pin_amt_install.pl -m** to refresh the data dictionary.

B

Account Migration Flags

This appendix describes the flags used in account migration in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About Job Status Flags
- About Batch Status Flags
- About Group Status Flags

About Job Status Flags

The AMM software sets jobs to a status listed in Table B-1. You can see a job's status by running a **list_jobs** report. See "Monitoring Job Status".

Table B-1   Job Status Flags

Status	Description
DISABLED	The job has been submitted but not enabled.
NOT_PROCESSED	The account migration job is enabled and waiting in the queue to be processed.
PRE_MIGRATION_WAITING	AMM is notifying ECE to suspend events for all accounts in the job.
PRE_MIGRATION	ECE acknowledged that it is suspending events for all accounts in the job. AMM is waiting a specified amount of time before starting migration.
READY	The job is ready to be processed.
IN_PROGRESS	The job is being processed by the AMM Controller.
FAILED	The job has been terminated.
BAL_MIGRATED	The account discount balance migrated successfully.
POST_MIGRATION_WAITING	AMM is notifying ECE that the job migrated successfully.
POST_MIGRATION	ECE acknowledged that it updated all account information.
PRE_JOB_RECYCLE	AMM is notifying ECE to resume processing events for accounts in the job.
JOB_RECYCLE	ECE acknowledged that it is ready to begin reprocessing events for accounts in the job.
FINISHED	The job has completed successfully.

About Batch Status Flags

AMM sets account batches to a status listed in Table B-2. You can check a batch's status by running a **job_details** report. See "Checking Job Details".

Table B-2   Batch Status Flags

Status	Description
NOT_PROCESSED	The account batch has not yet been migrated.
IN_PROGRESS	The account batch is currently being migrated.
FAILED	The account batch failed to migrate. All changes to the database have been rolled back.
FINISHED	The account batch migrated successfully. All changes to the database have been committed.

About Group Status Flags

AMM sets account groups to a status listed in Table B-3. You can check an account group's status by running a **group_details** report. See "Checking Account Group Details".

Table B-3   Group Status Flags

Status	Description
NOT_PROCESSED	The account group has not yet been migrated.
GROUP_DISABLING	All account group members are being disabled in the source database schema. That is, AMM is marking all account group members as invalid to prevent applications from accessing those accounts.
FAILED	AMM did not disable all account group members in the source schema.
GROUP_READY	All account group members were successfully disabled in the source schema. AMM can begin processing batches.
GROUP_IN_PROGRESS	The account group is currently being migrated.
GROUP_FAILED	The account group failed to migrate to the destination schema.
GROUP_FINISHED	The account group migrated successfully.

C

AMM Entity Relationship Diagram

This appendix describes the entity relationship diagram for the Oracle Communications Billing and Revenue Management (BRM) Account Migration Manager (AMM).

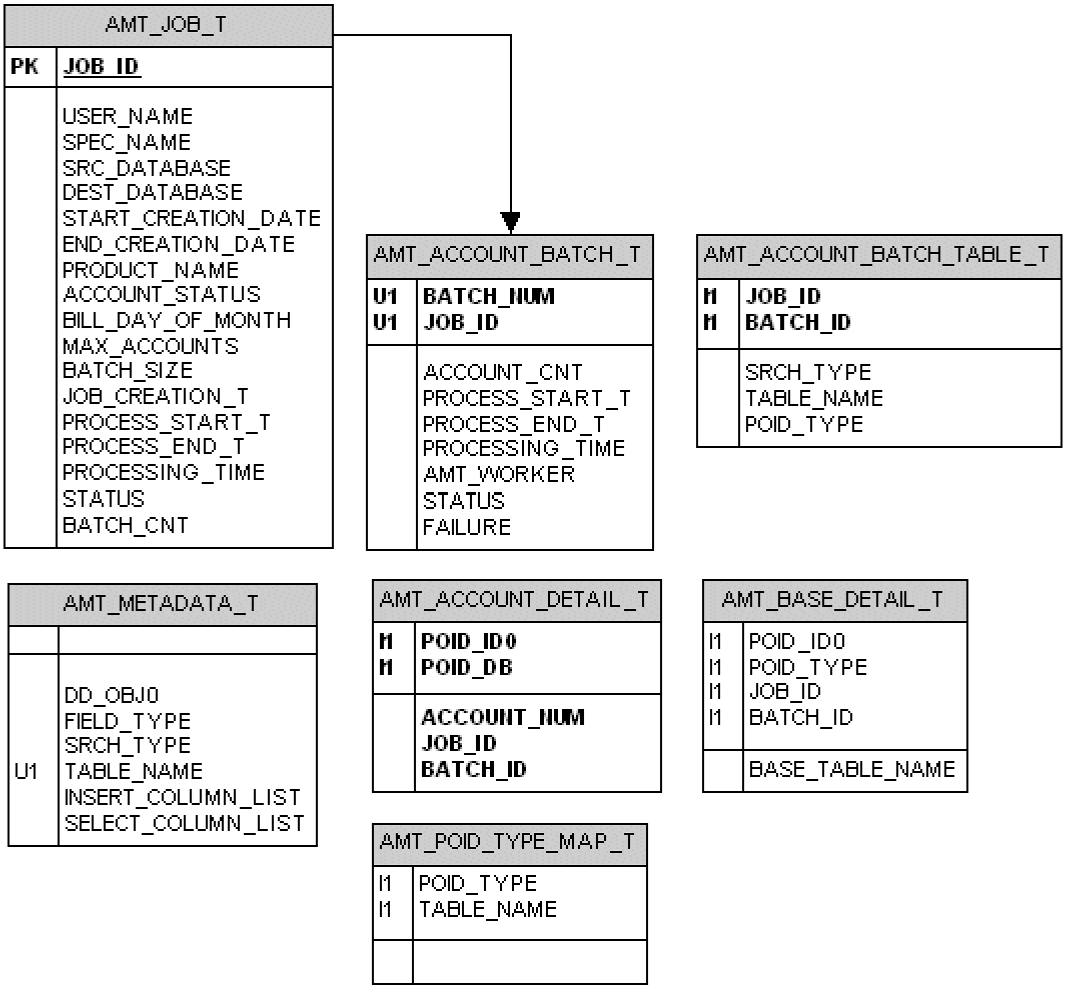
Topics in this document:

- AMM Entity Relationship Diagram

AMM Entity Relationship Diagram

Figure C-1 shows the Account Migration Manager Entity Relationship (ER) Diagram.

Figure C-1   Account Migration Manager ER Diagram


AMT_JOB_T
PK
JOB_ID
USER_NAME
SPEC_NAME
SRC_DATABASE
DEST_DATABASE
START_CREATION_DATE
END_CREATION_DATE
PRODUCT_NAME
ACCOUNT_STATUS
BILL_DAY_OF_MONTH
MAX_ACCOUNTS
BATCH_SIZE
JOB_CREATION_T
PROCESS_START_T
PROCESS_END_T
PROCESSING_TIME
STATUS
BATCH_CNT
AMT_ACCOUNT_BATCH_T
U1
BATCH_NUM
U1
JOB_ID
ACCOUNT_CNT
PROCESS_START_T
PROCESS_END_T
PROCESSING_TIME
AMT_WORKER
STATUS
FAILURE
AMT_ACCOUNT_BATCH_TABLE_T
I1
JOB_ID
I1
BATCH_ID
SRCH_TYPE
TABLE_NAME
POID_TYPE
AMT_METADATA_T
DD_OBJ0
FIELD_TYPE
SRCH_TYPE
U1
TABLE_NAME
INSERT_COLUMN_LIST
SELECT_COLUMN_LIST
AMT_ACCOUNT_DETAIL_T
I1
POID_ID0
I1
POID_DB
ACCOUNT_NUM
JOB_ID
BATCH_ID
AMT_BASE_DETAIL_T
I1
POID_ID0
I1
POID_TYPE
I1
JOB_ID
I1
BATCH_ID
BASE_TABLE_NAME
AMT_POID_TYPE_MAP_T
I1
POID_TYPE
I1
TABLE_NAME