## Contents

### Preface

- Audience .......................................................... ix
- Downloading Oracle Communications Documentation ............................................ ix
- Documentation Accessibility .................................................................................. ix
- Document Revision History .................................................................................. ix

### Part I  About Upgrading

#### 1 About Upgrading BRM Releases

- About Upgrading BRM to a New Release ................................................................. 1-1
- Planning Your Upgrade ......................................................................................... 1-2
  - Identifying Your Upgrade Team ........................................................................ 1-2
  - Identifying Who Is Affected by the Upgrade ....................................................... 1-2
  - Collecting Information about Your System ....................................................... 1-3
  - Determining the Impact of New Features ......................................................... 1-3
  - Estimating How Long the Upgrade Will Take .................................................... 1-4
    - How Long Will It Take to Run the Database Upgrade Scripts? .......................... 1-4
    - How Long Will It Take to Plan, Prepare for, Test, and Perform the Upgrade? ........ 1-4
- Updating Your System Environment ................................................................... 1-4
- Creating Test Environments .................................................................................. 1-5
  - Old Baseline Release ....................................................................................... 1-5
  - Old Customized Release .................................................................................. 1-6
- New Baseline Release ......................................................................................... 1-6
- New Customized Release (Test System) ............................................................... 1-6
- Transferring Customizations to the New Release .............................................. 1-6
  - Upgrading Customized Policy Source Files ...................................................... 1-7
  - Updating Configuration Files ........................................................................... 1-7
  - Updating Database Customizations .................................................................. 1-7
    - Modifying the Content of Custom Tables ....................................................... 1-7
    - Modifying the Structure of Custom Tables .................................................... 1-8
    - Fixing Standard Database Objects with Nonstandard Object IDs ................. 1-8
  - Updating Custom Reports ................................................................................ 1-8
  - Updating Custom Applications .......................................................................... 1-8
- Testing Your Upgraded System ........................................................................... 1-8
  - Running Old and New Versions in Parallel .................................................... 1-9
Part II Upgrade Impacts From 7.5 to 12.0

2 Feature Changes from BRM 7.5 to BRM 12.0
BRM Clients ........................................................................................................................................ 2-1
  BRM Clients and Billing Care.............................................................................................................. 2-1
  Brand Manager ................................................................................................................................ 2-1
  Pricing Center .................................................................................................................................. 2-1
Changes to Supported Software Versions ....................................................................................... 2-2
Content Manager ................................................................................................................................ 2-2
  Content SDK Removed from Content Manager ............................................................................. 2-2
Optional Managers .............................................................................................................................. 2-2
Service Managers ............................................................................................................................... 2-2
  AAA Managers Not Supported in BRM 12.0..................................................................................... 2-2
Voucher Manager ............................................................................................................................... 2-3
  Voucher Manager Support in BRM 12.0............................................................................................ 2-3
Web Services Manager ....................................................................................................................... 2-3
  Oracle GlassFish Server Not Supported in BRM 12.0................................................................. 2-3
System Administration ........................................................................................................................ 2-3
  French Direct Debit Not Supported in 12.0.................................................................................... 2-3
  PCM SDK is Now Part of BRM SDK .............................................................................................. 2-3
  SNMP Utilities Not Supported in 12.0 ............................................................................................ 2-3
  Storing Passwords in Oracle Wallet ............................................................................................... 2-3

3 Storable Class Changes from BRM 7.5 to BRM 12.0
New Storable Classes ....................................................................................................................... 3-1
Changed Storable Classes .................................................................................................................. 3-1

4 Opcode Changes from BRM 7.5 to BRM 12.0
New Opcodes ..................................................................................................................................... 4-1
Changed Opcodes ............................................................................................................................. 4-1
Obsoleted Policy Opcodes ............................................................................................................... 4-1
Obsoleted Standard Opcodes ......................................................................................................... 4-2

5 Utility Changes from BRM 7.5 to BRM 12.0
New Utilities .................................................................................................................................... 5-1
Changed Utilities ............................................................................................................................ 5-1

6 Notification Event Changes from BRM 7.5 to BRM 12.0
New Notification Events ................................................................................................................. 6-1
7 Module and iScript Changes from BRM 7.5 to 12.0

New Modules ................................................................. 7-1
Changed Modules .................................................................. 7-1
New iScripts ........................................................................... 7-1
Changed iScripts ................................................................... 7-1

Part III Upgrade Procedures

8 Upgrading BRM and Pipeline Manager

About the Upgrade................................................................. 8-1
Tasks Involved in the BRM 7.5-to-12.0 Upgrade Process ............... 8-1
Preparing for the Upgrade to BRM 12.0 or 12.0 Patch Sets .......... 8-2
  Shutting Down the Current Instance ...................................... 8-2
  Turning Off BRM Service Authentication and Authorization ... 8-3
Creating a Complete Backup of Your BRM 7.5 Data .................. 8-3
  Backing Up BRM Files ....................................................... 8-3
  Backing Up Your BRM 7.5 Database ................................. 8-4
Creating Your BRM 12.0 Database ....................................... 8-4
Upgrading to BRM 12.0 or 12.0 Patch Sets .......................... 8-4
  Importing Your BRM 7.5 Data ........................................... 8-4
  Installing Java ..................................................................... 8-5
  Installing Perl on Linux .................................................... 8-5
  Installing Perl on Solaris .................................................... 8-6
  Obtaining ojdbc JAR File ................................................. 8-7
  Setting the Environment Variables ..................................... 8-7
  Setting BRM Wallet Location in sqlnet.ora ......................... 8-8
Obtaining the BRM Software ............................................. 8-8
  Obtaining the BRM 12.0 Software ..................................... 8-8
  Obtaining the BRM 12.0 Patch Set Software ....................... 8-9
Installing BRM and Pipeline Manager for Upgrade .................. 8-9
Upgrading Your Data to BRM 12.0 ..................................... 8-12
  Pointing the BRM 12.0 Installation to the New Database Schema 8-12
Upgrading the BRM Database Schema to the BRM 12.0 Schema ... 8-12
  To upgrade the schema on single-schema systems: ............... 8-13
  Upgrading the Schema on Multischema Systems ................. 8-14
Upgrading the Pipeline Manager Database Schema to the BRM 12.0 Schema ........................................ 8-17
Completing the Upgrade to BRM 12.0 or 12.0 Patch Sets ........... 8-18
  Installing the BRM 12.0 Client Applications ........................ 8-18
  Adding Optional Components ......................................... 8-18
Updating Configurations .................................................. 8-18
  Update DM_IFW_SYNC Queue Database Configuration ........ 8-18
  Verify IFW_SYNC_QUEUE Queue in the Database ............. 8-19
  Update the Default Settings for the pin_rel Utility ............... 8-19
  Updating the Settings for the pin_virtual_time Utility .......... 8-19
Part IV  Migrating Pricing Data from The BRM Database to PDC

9  Migrating Pricing Data to PDC With ECE

About Migrating BRM Pricing Data to PDC With ECE ................................................................. 9-1
  About the BRM Data Analysis Phase for Migrating BRM Pricing Data to PDC With ECE........ 9-3
  About the Data Migration Phase for Migrating BRM Pricing Data to PDC With ECE .......... 9-3
Performing Migration of BRM Pricing Data to PDC With ECE .................................................... 9-5
  Creating a BackUp of Your BRM Database .............................................................................. 9-5
  Setting Up the BRM Development System ............................................................................ 9-5
    Configuring the Migration Utility ....................................................................................... 9-7
  Migrating the BRM Pricing Data to PDC With ECE ............................................................... 9-13
    Migrating BRM Configuration Objects ............................................................................ 9-13
    Migrating BRM Pricing Objects ....................................................................................... 9-14
About Migrating PDC Pricing Data to PDC With ECE ................................................................. 9-20
  About the Data Migration Phase for Migrating PDC Pricing Data to PDC With ECE ........ 9-21
Performing Migration of PDC Pricing Data to PDC With ECE .................................................... 9-22
  Creating a BackUp of Your PDC Database .......................................................................... 9-22
  Setting Up the PDC Development System .......................................................................... 9-23
  Migrating the PDC Pricing Data to PDC With ECE ............................................................... 9-24
    Migrating PDC Configuration Objects ............................................................................ 9-24
    Migrating PDC Pricing Objects ....................................................................................... 9-24
Fixing Errors Found in the Analysis Phase .................................................................................... 9-28
  Fixing Analysis "User Input Required" Errors ....................................................................... 9-28
    Identifying Resource Id for Non-Currency Resources Used in Batch Rating ................... 9-29
    Providing Values List for Expressions in Discount and Chargeshare Configurations ...... 9-29
Analysis Errors Requiring Pricing Data Changes ........................................................................ 9-29
  Multiple Impact Categories in a Rate Plan ......................................................................... 9-30
  Derived Impact Category Used in Zone Model .................................................................... 9-32
  Zone Items Differentiated Only by Service Code ............................................................... 9-33
  Extended Service Class Event Mapping ............................................................................. 9-33
  Service Contains Deal Mapped to Parent Service ............................................................... 9-34
  Using Overlapping Date Ranges to Sequentially Impact Multiple Resources .................. 9-34
  Quantity Brackets in a Fold ............................................................................................... 9-35
  BRM Object Names Containing Special Characters ............................................................. 9-35
10 Migrating Pricing Data to PDC With RRE and BRE

About Migrating BRM Pricing Data to PDC With RRE and BRE ................................................................. 10-1
  About the Data Analysis Phase for Migrating BRM Pricing Data to PDC With RRE and BRE .......... 10-3
  About the Data Migration Phase for Migrating BRM Pricing Data to PDC With RRE and BRE ... 10-3

Performing Migration of BRM Pricing Data to PDC With RRE and BRE .................................................. 10-4
Creating a BackUp of Your BRM Database ........................................................................................................ 10-5

Setting Up the BRM Development System ...................................................................................................... 10-5
  Configuring the Migration Utility .................................................................................................................... 10-7

Migrating the BRM Pricing Data to PDC With RRE and BRE ....................................................................... 10-12
  Migrating Configuration Objects .................................................................................................................. 10-13
  Migrating Pricing Objects ............................................................................................................................ 10-13

Fixing Errors Found in the Analysis Phase ....................................................................................................... 10-17
  Fixing Analysis "User Input Required" Errors .............................................................................................. 10-17
    Identifying Resource Id for Non-Currency Resources Used in Batch Rating ......................................... 10-17
    Providing Values List for Expressions in Discount and Chargeshare Configurations ............................. 10-17

Analysis Errors Requiring Pricing Data Changes ............................................................................................ 10-18
  Multiple Impact Categories in a Rate Plan .................................................................................................... 10-18
  Derived Impact Category Used in Zone Model ............................................................................................ 10-21
  Zone Items Differentiated Only by Service Code ......................................................................................... 10-21
  Extended Service Class Event Mapping ...................................................................................................... 10-21
  Service Contains Deal Mapped to Parent Service ....................................................................................... 10-22
  Using Overlapping Date Ranges to Sequentially Impact Multiple Resources ........................................ 10-22
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity Brackets in a Fold</td>
<td>10-23</td>
</tr>
<tr>
<td>BRM Object Names Containing Special Characters</td>
<td>10-23</td>
</tr>
<tr>
<td>Custom Fields</td>
<td>10-23</td>
</tr>
<tr>
<td><strong>Changes Required After Migration</strong></td>
<td>10-24</td>
</tr>
<tr>
<td>Selector with Multiple RUMs</td>
<td>10-24</td>
</tr>
<tr>
<td>Non-Currency Resources Used as Counters</td>
<td>10-24</td>
</tr>
<tr>
<td>BRM Rate Plans with Multiple Currencies</td>
<td>10-25</td>
</tr>
<tr>
<td><strong>Handling Migration Process Errors</strong></td>
<td>10-26</td>
</tr>
<tr>
<td>Restarting the Systems</td>
<td>10-27</td>
</tr>
<tr>
<td>Testing the Migrated Data</td>
<td>10-27</td>
</tr>
<tr>
<td>Importing the Migrated Data to the Production Systems</td>
<td>10-27</td>
</tr>
<tr>
<td>About Working with the Pricing Data after Migration</td>
<td>10-30</td>
</tr>
<tr>
<td><strong>11 Migration Reports</strong></td>
<td>11-1</td>
</tr>
<tr>
<td>About Migration Reports</td>
<td>11-1</td>
</tr>
<tr>
<td>Analysis Reports</td>
<td>11-1</td>
</tr>
<tr>
<td>Compatibility Reports</td>
<td>11-2</td>
</tr>
<tr>
<td>Migration Report</td>
<td>11-3</td>
</tr>
<tr>
<td>Process Report</td>
<td>11-3</td>
</tr>
<tr>
<td>Reconciliation Reports</td>
<td>11-3</td>
</tr>
<tr>
<td>Status Report</td>
<td>11-3</td>
</tr>
<tr>
<td><strong>12 BRM Pricing Migration Utility</strong></td>
<td>12-2</td>
</tr>
<tr>
<td>MigrateBRMPricing</td>
<td>12-2</td>
</tr>
</tbody>
</table>
Preface

This guide describes how to upgrade your existing system to later versions of Oracle Communications Billing and Revenue Management (BRM).

Audience

This guide is intended for system integrators, system administrators, database administrators, and others who install and upgrade BRM from release to release.

Downloading Oracle Communications Documentation

Product documentation is located on Oracle Help Center:

http://docs.oracle.com

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

https://edelivery.oracle.com

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at


Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Document Revision History

The following table lists the revision history for this book:

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E51037-01</td>
<td>December 2017</td>
<td>Initial release.</td>
</tr>
<tr>
<td>Version</td>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| E51037-02 | March 2018 | Documentation updates.  
|           |            | - Added the following sections:  
|           |            |   * Installing Java  
|           |            |   * Obtaining ojdbc JAR File  
|           |            | - Updated the following sections:  
|           |            |   * Installing Perl on Linux  
|           |            |   * Installing Perl on Solaris |
| E51037-03 | July 2018  | Documentation updates.  
|           |            | - Updated the note in "About Upgrading BRM Releases". |
| E51037-04 | September 2018 | Documentation updates.  
|           |            | - Updated the note in "Upgrading BRM and Pipeline Manager". |
| E51037-05 | September 2019 | Documentation updates for BRM 12.0 Patch Set 2.  
|           |            | - Updated the note in "About Upgrading BRM Releases".  
|           |            | - Updated the following sections:  
|           |            |   * BRM Clients  
|           |            |   * Upgrading BRM and Pipeline Manager |
This part contains a general overview about upgrading your Oracle Communications Billing and Revenue Management (BRM) software.

Part I contains the following chapter:

- About Upgrading BRM Releases
About Upgrading BRM Releases

This chapter provides general information on how to upgrade your existing system to the latest Oracle Communications Billing and Revenue Management (BRM) release.

In this document, the BRM release running on your production system is called the *old* release. The release you are upgrading to is called the *new* release. For example, if you are upgrading from BRM 7.5 to BRM 12.0, BRM 7.5 is the old release, and BRM 12.0 is the new release.

**Important:** A direct upgrade from the BRM 7.4 release is not supported. If you are upgrading from BRM 7.4, you must upgrade your system to BRM 7.5 first.

For information on the BRM 7.5 releases from which you can directly upgrade to BRM 12.0 or 12.0 patch sets, see the discussion about supported upgrades in *BRM Compatibility Matrix*.

### About Upgrading BRM to a New Release

Upgrading to a new release is a four-part process:

1. Plan the upgrade process.
2. Implement and test the upgrade on a test system.
3. Prepare to upgrade your production system.
4. Implement and test the upgrade on the production system.

The upgrade process includes these tasks:

- **Install BRM 12.0 or a BRM 12.0 patch set** (without installing the database schema).

- **Update the BRM database**. The new BRM release includes an updated database schema with new tables and indexes. You use upgrade scripts to update your BRM database to the new schema.

- **Reimplement customizations**.
  - Source code for policy opcodes can change in a new release. You must merge your old release customizations into the new policy source code and recompile the policy Foams.
  - To support new functionality, the new software includes new configuration files. You must update those files to include the customizations made to your old system.
Other customizations in your system; such as customized invoicing, reports, general ledger reporting, and client applications; might need to be updated to work with the new BRM software.

- **Implement new features.** You can implement new BRM functionality to improve your BRM system. See the information about new features in BRM Release Notes.

The basic steps for upgrading are:

1. Back up files.
2. Turn off service authentication and authorization.
3. Shut down the old release.
4. Back up the old release’s database.
5. Install BRM 12.0 or a BRM 12.0 patch set (without installing the database schema).
6. Upgrade the BRM database schema.
7. Install BRM 12.0 client applications and optional components.
8. Add customizations.
9. Restore service authentication.

---

**Note:** There are additional steps if you use a multischema system, and optional steps for loading data.

---

### Planning Your Upgrade

To plan your upgrade, you can perform the following tasks:

1. **Identifying Your Upgrade Team**
2. **Identifying Who Is Affected by the Upgrade**
3. **Collecting Information about Your System**
4. **Determining the Impact of New Features**
5. **Estimating How Long the Upgrade Will Take**

#### Identifying Your Upgrade Team

Your upgrade team should include the following team members:

- A database administrator to manage the database upgrade and tune the database.
- A system administrator to manage the hardware and system architecture.
- A business analyst to make business decisions about changes to your BRM implementation.
- A customer service representative (CSR) supervisor to assess the impact on CSRs.

#### Identifying Who Is Affected by the Upgrade

You should identify who might be affected by the upgrade. For example:

- You might need to give your customers advance notice of any system downtime.
- Tell your system administrators in advance about any changes to the system architecture.
Planning Your Upgrade

About Upgrading BRM Releases

- Notify Oracle so that we can help you anticipate and avoid problems. Technical support might have additional information about upgrading BRM or information specific to your implementation.

Collecting Information about Your System

When you upgrade, you must know all the customizations you implemented in the old release. To prepare for the upgrade, find or create the following documents:

- **Implementation design documents.** When you first implemented BRM, you should have created documents that explained your business requirements and the customizations you made to meet those requirements. These documents help you create a list of customized components. You can also compare these documents with the documentation on new BRM features to find out whether any of your customizations can now be implemented by using standard BRM functionality.

- **List of customized components.** This should list every file created or modified for the original implementation. You need this list to know which files must be checked against the new release for changes.

  Customizations might include the following:
  - Additional Data Managers (DMs)
  - Custom Facility Modules (FMs)
  - Custom client applications
  - Custom DLLs
  - Custom reports
  - Merging container DESC.dsc files
  - Modified storable classes
  - Additional storable classes
  - Custom table indexes
  - Modified configuration, properties, and INI files
  - Custom Web pages
  - A gateway service that provides access to a legacy system

Determining the Impact of New Features

You might need to make changes to your current system to accommodate new functionality in the new release. For example, if the new release changes how balance impact rounding works, you might need to modify your pricing components to support the new rounding method.

The following features are typically affected by new releases:

- Product offerings (rating)
- Billing and invoicing
- General ledger (G/L IDs)
- Web pages
- BRM reports
Updating Your System Environment

- Client applications used by CSRs
- Components that integrate credit card processors and tax software
- Discount balancing

In addition, new features might include default functionality that you implemented as a customization. In that case, it is best to replace your customizations with the new feature.

Estimating How Long the Upgrade Will Take

When estimating the time it will take to upgrade, consider the following:

- How Long Will It Take to Run the Database Upgrade Scripts?
- How Long Will It Take to Plan, Prepare for, Test, and Perform the Upgrade?

How Long Will It Take to Run the Database Upgrade Scripts?

This is an important consideration because services might be suspended and authentication and authorization might be unavailable while you upgrade the database.

The best way to determine how long the database upgrade will take is to run the upgrade scripts on a test system that duplicates the data in your production system (see "Creating Test Environments").

In general, it takes longer to upgrade large databases with large tables. A large database can take from 8 to 48 hours to upgrade.

Reducing BRM System Downtime by Purging or Archiving Old Data

The upgrade scripts convert your old release data to the new release format. The time required to complete an upgrade is directly proportional to the size of your database. To save time, purge or archive data that is no longer required before you shut down your production system to perform the upgrade.

Because event tables consume most of the space in a database, you can significantly reduce the size of the database by purging unneeded event objects. If you cannot purge event objects, archive those that are no longer needed.

How Long Will It Take to Plan, Prepare for, Test, and Perform the Upgrade?

Depending on the size and complexity of your BRM implementation, the entire upgrade process can take several months. If your system architecture and customization documentation is complete and up-to-date, the time is significantly shorter.

For help with your upgrade, contact Oracle Support.

Updating Your System Environment

Before upgrading, prepare your system environment:

- Install the latest releases.

Install the latest BRM-supported release of your operating system and database software. Include the latest patches. If you are not running the latest supported release of Oracle, you might need to upgrade your database software before upgrading BRM. For more information, see the following:
Creating Test Environments

To test your upgrade, create the environments described in this section. You use these environments to do the following:

- Compare the default behavior of the old and new releases.
- Determine what customizations you made in the old release.
- Test the upgrade process and its results.

**Tip:**
- You can install multiple BRM instances on a single UNIX machine. See “Installing and Configuring Multiple Instances of BRM on One Machine” in BRM Installation Guide.
- If you install BRM on multiple systems, you can save time by compressing and moving folders and files instead of running the BRM installer on each system. When you copy the files to other systems, you might need to change some configuration file values, such as port numbers, manually if you do not use the automated installer.

**Old Baseline Release**

Your *old baseline release* system should run the old BRM release with the latest ServicePak but without any customizations. Use this system to do the following:

- Determine what default behavior in the old release has changed in the new release by comparing the old baseline release with the "New Baseline Release".
- Determine what customizations you made in the old release by comparing the old baseline release with your existing system, which this document calls the "Old Customized Release".

Creating Test Environments

About Upgrading BRM Releases 1-5
Old Customized Release

Your old customized release system should run the old BRM release with your customizations. This system should be identical to your current production system. To ensure that your test upgrade ("New Customized Release (Test System)") is working properly, compare the behavior of the new customized release with the old customized release.

New Baseline Release

Your new baseline release system should run the new BRM release without any customizations. Use this system to find out how the latest BRM release works.

New Customized Release (Test System)

Your new customized release (or test) system should run the new BRM release with your customizations and an upgraded BRM database. Use this system to test the following aspects of the upgrade:

- **The upgrade process.** The procedures used to create this system should be as close as possible to the procedures used to upgrade your production system. This ensures that you test such tasks as turning authentication on and off and copying over configuration files.

  For information on the standard procedure for upgrading a production system, see "Upgrading Your Production System".

**Tip:** To create this system, perform a test upgrade on your "Old Customized Release".

- **The upgrade results.** To test the outcome of your upgrade process, run all tests on this system. See "Testing Your Upgraded System".

**Tip:** This system’s database should include the same data as your "Old Customized Release" database. That way, you can run tests on the same data and compare the results.

Transferring Customizations to the New Release

This section explains how to transfer customizations from the old release to the new release:

- Upgrading Customized Policy Source Files
- Updating Configuration Files
- Updating Database Customizations
- Updating Custom Reports
- Updating Custom Applications
Tip: When you upgrade to a new BRM release, you first install the default BRM server configuration, and you then install your customized files. To ensure that you add all your customized files to the new release, create a package that includes all your customized files.

Upgrading Customized Policy Source Files

New releases often include changes to policy opcodes. If you customized your policy opcodes, you might need to re-create your customizations after you install the new BRM release:

1. See your internal documentation to find out what customizations were made to policy source files. If the customizations were not documented, use a diff tool to compare the source code in the "Old Baseline Release" with the source code in your "Old Customized Release". Source code is stored in the BRM_home/source folder.

2. When you know what your customizations are, use a diff tool to compare your customized source code with the source code in the "New Baseline Release".

3. Determine whether new features implement any of your customizations or make them irrelevant.

4. Merge the policy source file customizations that you still need into the new release source code. As you do so, find any changes to input and output flists. It is common for fields to change or to change from required to optional (or vice versa).

5. Using the libraries in the new release, recompile your custom code in the new release.

6. Test the new source code by using the functionality that it customizes.

Updating Configuration Files

When you install a new BRM release, you install new configuration files, such as the Connection Manager (CM) pin.conf file. You must update these files to include the customizations you made to them in your old system. For more information, see “Using Configuration Files to Connect and Configure Components” in BRM System Administrator’s Guide.

Updating Database Customizations

If you added custom storable classes to your old release, corresponding custom tables were added to your BRM database, and corresponding custom definitions were added to your BRM data dictionary. These customizations are not modified by the database upgrade scripts.

Modifying the Content of Custom Tables

Depending on how the new release differs from the old release, you might need to modify the old data in your custom tables to accommodate the new release. For example, a custom table might store phone numbers in the following format: 408-555-1212. Opcodes in the new release, however, might need a different format, such as 1-408-555-1212.
Modifying the Structure of Custom Tables

Depending on the changes introduced by the new release, you might need to modify the structure of your custom database tables. For example, you might need to add, delete, or change the size of a column. To make such changes, use either Developer Center or the `pin_deploy` utility after running the database upgrade scripts. Those tools will automatically update your new database schema and data dictionary.

Fixing Standard Database Objects with Nonstandard Object IDs

When you run the database upgrade scripts, the scripts overwrite standard database objects. If you deleted standard objects and re-created them with nonstandard object IDs when customizing BRM, the upgrade scripts delete those objects.

To drop a standard object from the upgraded database and re-create the object with a nonstandard ID, use the `testnap create 1 poid` command. This command enables you to use the POID in the input file to re-create the object instead of using a value from the POID sequence.

Updating Custom Reports

Since BRM reports read data stored in objects, changes to the database schema often require changes to reports. Sometimes it is easier to design new reports to work with the new database schema than to update old reports, especially when there are large-scale schema changes.

For more information, see “About BRM Reports” in BRM Reports.

Updating Custom Applications

Use your "New Customized Release (Test System)" to test all custom applications that call opcodes or manipulate data in the database. Changes to storable classes and opcodes in the new release might make such applications function differently.

**Important:** You must recompile your custom applications with the new BRM libraries.

Testing Your Upgraded System

To test your "New Customized Release (Test System)", perform the tests listed in "Testing Checklist". When testing, cover all aspects of the system, including CSR activity, customer logins and service usage, and billing.
When you have completed all your tests without finding any errors, run all the tests again, twice. You should run through the tests twice without error before considering the test cycle complete.

If you find any problems that indicate a BRM problem, submit it to Oracle.

---

**Tip:** Create an upgrade process document that includes a checklist of the upgrade tasks. Part of your test is to ensure that the checklist is complete because you will use it when upgrading your production system.

---

**Important:** Document all customizations you make during the upgrade. You will need to know about them the next time you upgrade.

---

**Running Old and New Versions in Parallel**

A good way to test your upgraded BRM implementation is to run the old release and the new release in parallel. To do so, run the old system as your production system, and import all your data into the new system. You can then run billing and perform other tasks on both systems and compare the results.

---

**Testing the BRM Database**

When you install a new BRM release, you run upgrade scripts to update the database schema. The scripts update the database tables and fields, including the BRM database definition. The scripts modify only default BRM objects and do not affect custom objects.

You should load actual production data into the test environment to run the database upgrade scripts with your real information. If you do not have enough hardware to load the entire database, include at least some account information.

When running the upgrade scripts on your test system, look for the following:

- Data errors in your current database
- Custom tables and fields that are not being used

---

**Troubleshooting Your Upgraded System**

For information about BRM error messages and other problems that you encounter in your upgraded system, see the following documents:

- “Reference Guide to BRM Error Codes” in *BRM System Administrator’s Guide*
- “Resolving Problems in Your BRM System” in *BRM System Administrator’s Guide*

Upgrading BRM can expose implementation problems. While running tests, some problems you find might be caused in part by the following:

- **Missing data.** If your initial BRM implementation included the conversion and migration of legacy data, the conversion might not have created all necessary objects or fully populated all fields. In addition, data is sometimes mistakenly deleted. The missing data might not affect the existing BRM implementation, but later versions of BRM might need it.
Preparing for the Production System Upgrade

■ **Undocumented customizations.** If you find what appears to be an undocumented customization, compare the "Old Baseline Release" with your "Old Customized Release". If you find an undocumented customization, be sure to document it!

■ **New functionality.** BRM functionality changes between releases. These changes might enable or require you to get rid of or change some of your customizations.

**Testing Checklist**

You should have a list of tests that were performed during your initial BRM implementation. In addition to running all those tests, you might run new tests to cover new functionality.

---

**Important:** Before running tests in a *production* environment, ensure that all entries for the `pin_virtual_time` utility were removed from configuration files. If you are running in a *test* environment, it is not necessary to remove entries for the `pin_virtual_time` utility.

---

These are the basic tests you should perform on your test system:

■ Create accounts using the client applications.

■ Generate and rate usage events.

■ Run billing, including requesting and receiving payments. Check all log files and invoices after running billing.

■ Run the `/pin_ledger_report` utility on your "Old Customized Release" and your test system, and compare the results. If the database on both systems includes the same data, the results should be the same.

■ Run all the client applications that your implementation uses.

■ Test your product offerings by committing them to the database and purchasing every package and bundle.

■ Test all your optional components, such as LDAP Manager and GSM Manager.

■ Test credit card processing, including a live connection to the credit card processor.

■ Test tax calculation.

■ Test all functionality that involves multiple currencies.

■ Run all BRM reports that you use. If possible, run the reports against the same data in your "Old Customized Release" and your test system.

■ Review all log files for warnings and errors.

■ Test all interfaces to external systems.

■ Develop and test a fallback plan in case you run into problems during the production system upgrade. For example, after you shut down your production system, you should create a full backup of the system in case you need to restore it.

**Preparing for the Production System Upgrade**

Before you upgrade your production system, do the following:

■ Make a backup of the data in your production system.
Ensure that all files required for customizing the system are available and ready to be copied to the upgraded system. This might include configuration files and Facility Modules (FMs). Test the procedure for copying these files to the upgraded system.

Be prepared to run a full integration test on your production system after the upgrade scripts have run. Prepare any test scripts and test them before shutting down BRM.

Inform anyone who needs to know about the upgrade. See "Identifying Who Is Affected by the Upgrade".

Create a checklist for the upgrade procedure. (You should have created one while testing the upgrade. See "Testing Checklist".)

Prepare a production staging system. This system includes the "New Baseline Release" with your customized files. You build this system after testing is complete. It serves as a clean system from which to copy files to your production system. To avoid resource contention, run this system on its own dedicated hardware to simulate production performance more accurately. If you have limited resources, you can use your "New Customized Release (Test System)" system as the production staging system.

**Upgrading Your Production System**

The procedures used to upgrade your production system should be almost identical to the procedures used to create your "New Customized Release (Test System)".

---

**Caution:** Do not begin your upgrade until you have backed up your production system.
This part contains information on the impacts of upgrades from Oracle Communications Billing and Revenue Management (BRM) Release 7.5 to Release 12.0.

This part contains the following chapters:

- Feature Changes from BRM 7.5 to BRM 12.0
- Storable Class Changes from BRM 7.5 to BRM 12.0
- Opcode Changes from BRM 7.5 to BRM 12.0
- Utility Changes from BRM 7.5 to BRM 12.0
- Notification Event Changes from BRM 7.5 to BRM 12.0
- Module and iScript Changes from BRM 7.5 to 12.0
This chapter provides upgrade impact information for Oracle Communications Billing and Revenue Management (BRM) Release 7.5 to BRM 12.0. It describes the feature changes that affect your BRM system and what you must consider when you upgrade from 7.5 to 12.0.

For information on planning your upgrade implementation, such as setting up your development and test environments, see "About Upgrading BRM Releases".

**BRM Clients**

The following are the changes made to the client applications in BRM 12.0.

**BRM Clients and Billing Care**

In BRM 12.0, the following client applications are supported only from BRM 12.0 Patch Set 2:

- Customer Center (including Customer Center SDK)
- GSM Manager Customer Center Extension
- Collections Center
- Payment Center
- Payment Tool

You can use these applications or Oracle Communications Billing and Revenue Management Billing Care for customer, payment management, and some collection tasks.

For more information, see the following documents:

- *BRM Concepts*
- Billing Care Online Help

**Brand Manager**

BRM 12.0 no longer supports Brand Manager for creating new brands.

**Pricing Center**

In BRM 12.0, Pricing Center is supported from BRM 12.0 Patch Set 2. You can use Pricing Center to create and maintain price lists if you are using Pipeline Manager for rating. If you are using Oracle Communications Elastic Charging Engine (ECE) for
rating, you must use Oracle Communications Pricing Design Center for creating and maintaining your price lists and other product offerings.

For more information, see the following documents:
- BRM Concepts
- BRM Creating Product Offerings

Changes to Supported Software Versions

Every release can include changes to supported software. Always read the information about software compatibility in the BRM Installation Guide.

Content Manager

The following are the changes made to Content Manager in BRM 12.0.

Content SDK Removed from Content Manager

BRM 12.0 no longer supports Content SDK and the AAA Opcodes associated with Content Manager. You must now use Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) instead for AAA functions.

For more information, see ECE Implementation Guide.

Optional Managers

BRM 12.0 no longer supports the following optional managers:
- Email Manager
- IMT Manager
- RADIUS Manager
  You must now use Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) RADIUS Gateway instead for authorization and authentication tasks. For more information, see ECE Implementation Guide.
- Resource Reservation Manager
  You must now use Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) to reserve customer account balance resources in advance of a network session. For more information, see ECE Implementation Guide.
- Vertex Quantum Manager

Service Managers

The following are the changes made to service managers in BRM 12.0.

AAA Managers Not Supported in BRM 12.0

BRM 12.0 no longer supports the following BRM service managers:
- GSM AAA Manager
- GPRS AAA Manager
- Services Framework AAA Manager
You must now use Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) RADIUS Gateway instead for processing AAA requests for BRM services.

For more information, see *ECE Implementation Guide*.

**Voucher Manager**

The following are the changes made to Voucher Manager in BRM 12.0.

**Voucher Manager Support in BRM 12.0**

BRM 12.0 supports Voucher Manager only if you have already installed it in your existing BRM system.

**Web Services Manager**

The following are the changes made to Web Services Manager in BRM 12.0.

**Oracle GlassFish Server Not Supported in BRM 12.0**

BRM 12.0 no longer supports Oracle GlassFish Server. You can install Web Services Manager only on the following servers:

- Oracle WebLogic Server
- Apache Tomcat server

**System Administration**

The following are the changes made to system administration in BRM 12.0.

**French Direct Debit Not Supported in 12.0**

BRM 12.0 no longer supports French Direct Debit payment processing.

**PCM SDK is Now Part of BRM SDK**

In BRM 12.0, PCM SDK is a part of BRM SDK and it is installed along with the BRM server. You cannot install PCM SDK independent of the BRM server.

**SNMP Utilities Not Supported in 12.0**

BRM 12.0 no longer supports SNMP utilities to monitor pipeline processes.

**Storing Passwords in Oracle Wallet**

In earlier releases, some sensitive information, such as database and account passwords, was stored in the `Infranet.properties` and `pin.conf` configuration files. Now, the BRM Installer by default stores sensitive information in the Oracle wallet and BRM components retrieve the passwords from the Oracle wallet. This ensures that the sensitive information is stored and accessed securely in BRM.
Note: Infranet.properties and pin.conf configuration files are supported for backward compatibility.

For more information on storing information in Oracle wallet, see the discussion about storing passwords in Oracle wallet in BRM System Administrator’s Guide.
This chapter provides upgrade impact information between Oracle Communications Billing and Revenue Management (BRM) Release 7.5 and BRM 12.0. It describes the storable class changes that affect your BRM system and what you must consider when you upgrade from BRM 7.5 to BRM 12.0. It also provides information about storable class index and schema changes.

See "About Upgrading BRM Releases" for information on planning your upgrade implementation, such as setting up your development and test environments.

**New Storable Classes**

No new storable classes were added in BRM 12.0.

**Changed Storable Classes**

No new storable classes were changed in BRM 12.0.
Opcode Changes from BRM 7.5 to BRM 12.0

This chapter describes the opcode changes that affect your Oracle Communications Billing and Revenue Management (BRM) system and what you must consider when you upgrade from BRM 7.5 to BRM 12.0.

See “About Upgrading BRM Releases” for information on planning your upgrade implementation, such as setting up your development and test environments.

New Opcodes

No new policy or standard opcodes were added in BRM 12.0.

Changed Opcodes

No policy or standard opcodes were changed in BRM 12.0.

Obsoleted Policy Opcodes

The following policy opcodes were obsoleted between BRM 7.5 and BRM 12.0.

- PCM_OPCONTENT_POLACCOUNTING
- PCM_OPCONTENT_POLAUTHORIZE
- PCM_OPCONTENT_POLPOSTACCOUNTING
- PCM_OPCONTENT_POLPOSTAUTHORIZE
- PCM_OGPGRS_AAA_POLACC_ON_OFF_SEARCH
- PCM_OGPGRS_AAA_POLAUTHORIZE
- PCM_OGPGRS_AAA_POLAUTHORIZED_PREP_INPUT
- PCM_OGPGRS_AAA_POLREAUTHORIZE_PREP_INPUT
- PCM_OGPGRS_AAA_POLSEARCH_SESSION
- PCM_OGPGRS_AAA_POL_STOP_ACCOUNTING_PREP_INPUT
- PCM_OGPGRS_AAA_POL_UPDATE_ACCOUNTING_PREP_INPUT
- PCM_OGPGRS_AAA_POLACC_ON_OFF_SEARCH
- PCM_OGPGRS_AAA_POLAUTHORIZE
- PCM_OGPGRS_AAA_POLAUTHORIZED_PREP_INPUT
- PCM_OGPGRS_AAA_POL_POST_PROCESS
Obsoleted Standard Opcodes

- PCM_OP_GSM_AAA_POL_REAUTHORIZE_PREP_INPUT
- PCM_OP_GSM_AAA_POL_SEARCH_SESSION
- PCM_OP_GSM_AAA_POL_STOP_ACCOUNTING_PREP_INPUT
- PCM_OP_GSM_AAA_POL_UPDATE_ACCOUNTING_PREP_INPUT
- PCM_OP_TCF_AAA_POL_GET_SUBSCRIBER_PROFILE
- PCM_OP_TCF_AAA_POL_POST_PROCESS
- PCM_OP_TCF_AAA_POL_VALIDATE_LIFECYCLE
- PCM_OP_TCF_AAA_POL_MATCH_CONTINUATION_CALL
- PCM_OP_TERM_POL_ACCOUNTING
- PCM_OP_TERM_POL_AUTHORIZE
- PCM_OP_TERM_POL_REVERSE_IP

Obsoleted Standard Opcodes

The following standard opcodes were obsoleted between BRM 7.5 and BRM 12.0.

- PCM_OP_ASM_CLOSE_ACTIVE_SESSION
- PCM_OP_ASM_CREATE_ACTIVE_SESSION
- PCM_OP_ASM_FIND_ACTIVE_SESSION
- PCM_OP_ASM_UPDATE_ACTIVE_SESSION
- PCM_OP_CONTENT_ACCOUNTING
- PCM_OP_CONTENT_AUTHENTICATE
- PCM_OP_CONTENT_AUTHORIZE
- PCM_OP_CONTENT_CANCEL_AUTHORIZATION
- PCM_OP_CONTENT_FIND
- PCM_OP_FILTER_SET_CREATE
- PCM_OP_FILTER_SET_DELETE
- PCM_OP_FILTER_SET_UPDATE
- PCM_OP_GSM_AAA_ACCOUNTING_OFF
- PCM_OP_GSM_AAA_ACCOUNTING_ON
- PCM_OP_GSM_AAA_AUTHENTICATE
- PCM_OP_GSM_AAA_AUTHORIZE
- PCM_OP_GSM_AAA_CANCEL_AUTHORIZATION
- PCM_OP_GSM_AAA_REAUTHORIZE
- PCM_OP_GSM_AAA_START_ACCOUNTING
- PCM_OP_GSM_AAA_STOP_ACCOUNTING
- PCM_OP_GSM_AAA_UPDATE_ACCOUNTING
- PCM_OP_GSM_AAA_UPDATE_AND_REAUTHORIZE
- PCM_OP_TCF_AAA_ACCOUNTING
- PCM_OP_TCF_AAA_ACCOUNTING_OFF
- PCM_OP_TCF_AAA_ACCOUNTING_ON
- PCM_OP_TCF_AAA_ACCOUNTING_PREP_INPUT
- PCM_OP_TCF_AAA_AUTHENTICATE
- PCM_OP_TCF_AAA_AUTHORIZE
- PCM_OP_TCF_AAA_AUTHORIZE_PREP_INPUT
- PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION
- PCM_OP_TCF_AAA_DETECTCONTINUATION_CALL
- PCM_OP_TCF_AAA_GET_SUBSCRIBER_PROFILE
- PCM_OP_TCF_AAA_QUERY_BALANCE
- PCM_OP_TCF_AAA_REAUTHORIZE
- PCM_OP_TCF_AAA_REAUTHORIZE_PREP_INPUT
- PCM_OP_TCF_AAA_REFUND
- PCM_OP_TCF_AAA_SEARCH_SESSION
- PCM_OP_TCF_AAA_SERVICE_PRICE_ENQUIRY
- PCM_OP_TCF_AAA_START_ACCOUNTING
- PCM_OP_TCF_AAA_STOP_ACCOUNTING
- PCM_OP_TCF_AAA_STOP_ACCOUNTING_PREP_INPUT
- PCM_OP_TCF_AAA_UPDATE_ACCOUNTING
- PCM_OP_TCF_AAA_UPDATE_ACCOUNTING_PREP_INPUT
- PCM_OP_TCF_AAA_VALIDATE_LIFECYCLE
- PCM_OP_TERM_IP_DIALUP_ACCOUNTING_OFF
- PCM_OP_TERM_IP_DIALUP_ACCOUNTING_ON
- PCM_OP_TERM_IP_DIALUP_AUTHENTICATE
- PCM_OP_TERM_IP_DIALUP_AUTHORIZE
- PCM_OP_TERM_IP_DIALUP_START_ACCOUNTING
- PCM_OP_TERM_IP_DIALUP_STOP_ACCOUNTING
- PCM_OP_TERM_IP_DIALUP_UPDATE_ACCOUNTING
Utility Changes from BRM 7.5 to BRM 12.0

This chapter provides upgrade impacts information for Oracle Communications Billing and Revenue Management (BRM) Release 7.5 to BRM 12.0. It describes the utility changes that affect your BRM system, and what you must consider when you upgrade from 7.5 to 12.0.

See "About Upgrading BRM Releases" for information on planning your upgrade implementation, such as setting up your development and test environments.

New Utilities

Table 5–1 lists the utilities that were added in BRM 12.0.

<table>
<thead>
<tr>
<th>New Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pin_create_server_cert</td>
<td>Generates Transport Layer Security (TLS) certificates for the BRM server processes.</td>
</tr>
<tr>
<td>pin_config_editor</td>
<td>Sets and gets configuration entries for JAVA PCM client applications from the client wallet.</td>
</tr>
</tbody>
</table>

Changed Utilities

Table 5–2 contains a list of the utilities that were changed in BRM 12.0.

<table>
<thead>
<tr>
<th>Changed Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pin_crypt_app</td>
<td>Modified to support only Oracle ZT PKI encryption algorithm for encryption and to store or retrieve configuration entries in the Oracle wallet.</td>
</tr>
</tbody>
</table>
This chapter provides upgrade impact information for Oracle Communications Billing and Revenue Management (BRM) 7.5 to BRM Release 12.0. It describes the notification event changes that affect your BRM system, and what you must consider when you upgrade from 7.5 to 12.0.

For information on planning your upgrade implementation, such as setting up your development and test environments, see "About Upgrading BRM Releases".

**New Notification Events**

No notification events were added in BRM 12.0.

**Changed Notification Events**

No notification events were changed in BRM 12.0.
Module and iScript Changes from BRM 7.5 to 12.0

This chapter provides upgrade impact information for Oracle Communications Billing and Revenue Management (BRM) 7.5 to BRM Release 12.0. It describes the module and iScript changes that affect your BRM system, and what you must consider when you upgrade from 7.5 to 12.0.

For information on planning your upgrade implementation, such as setting up your development and test environments, see "About Upgrading BRM Releases".

**New Modules**

No new modules were added in BRM 12.0.

**Changed Modules**

No modules were changed in BRM 12.0.

**New iScripts**

No new iScripts were added in BRM 12.0.

**Changed iScripts**

No iScripts were changed in BRM 12.0.
This part contains information on the upgrade procedures for different categories of Oracle Communications Billing and Revenue Management (BRM) software.

Part IV contains the following chapters:

- Upgrading BRM and Pipeline Manager
This chapter contains the procedures required to upgrade Oracle Communications Billing and Revenue Management (BRM) from 7.5 to 12.0 or 12.0 patch sets. It covers BRM on Linux and Solaris.

**Important:** When upgrading to BRM 12.0, note the following:

- A direct upgrade from the BRM 7.4 release is not supported. If you are upgrading from BRM 7.4, you must upgrade your system to BRM 7.5 first.

  For information on the BRM 7.5 releases from which you can directly upgrade to BRM 12.0 or 12.0 patch sets, see the discussion about supported upgrades in *BRM Compatibility Matrix*.

- Installing BRM on HP-UX IA64 and AIX platforms is not supported.

*Before* performing this upgrade, see "About Upgrading BRM Releases" for information on how to plan, prepare for, and test your upgrade.

**About the Upgrade**

This upgrade includes the changes that are necessary for upgrading BRM 7.5 to BRM 12.0 or 12.0 patch sets.

It upgrades the following components:

- BRM
- BRM software Development Kit (BRM SDK)
- Pipeline Manager
- Pipeline Portal Development Kit (Pipeline PDK)

**Tasks Involved in the BRM 7.5-to-12.0 Upgrade Process**

This section provides a list of the tasks required to update BRM 7.5 to BRM 12.0 or 12.0 patch sets. Some tasks are optional or apply only to certain platforms or system configurations. Be sure to check whether a task is required for your system.

**Caution:** When upgrading a multischema system, pay close attention to the system on which each task is performed.
The following tasks must be completed to upgrade your BRM system:

1. Shut down the current instance of BRM. See "Shutting Down the Current Instance".
2. Turn off service authentication and authorization. See "Turning Off BRM Service Authentication and Authorization".
3. Creating a complete backup of your BRM 7.5 data. See "Creating a Complete Backup of Your BRM 7.5 Data".
4. Create a new schema in BRM 12.0. See "Creating Your BRM 12.0 Database"
5. Import your data BRM 7.5 data. See "Importing Your BRM 7.5 Data"
6. Install Java. See "Installing Java".
7. Install Perl (32 bit).
   - For instructions on installing Perl on Linux, see "Installing Perl on Linux".
   - For instruction on installing Perl on Solaris, see "Installing Perl on Solaris".
8. Obtain the ojdbc8.jar jar file. See "Obtaining ojdbc JAR File".
9. Set the environment variables. See "Setting the Environment Variables".
10. Set the BRM wallet location in sqlnet.ora. See "Setting BRM Wallet Location in sqlnet.ora".
11. Install the upgrade package. See "Installing BRM and Pipeline Manager for Upgrade".
12. Point your BRM 12.0 installation to the new database schema. See "Pointing the BRM 12.0 Installation to the New Database Schema".
13. Upgrade the BRM (Portalbase) database schema to the BRM 12.0 schema. See "Upgrading the BRM Database Schema to the BRM 12.0 Schema".
14. Upgrade the Pipeline Manager database schema to the BRM 12.0 schema. See "Upgrading the Pipeline Manager Database Schema to the BRM 12.0 Schema."
15. Install BRM 12.0 client applications. See "Installing the BRM 12.0 Client Applications".
16. Add optional components. See "Adding Optional Components".
17. Check other database configurations. See "Updating Configurations".
18. Restore customizations in BRM 12.0. See "Restoring Customizations".

Preparing for the Upgrade to BRM 12.0 or 12.0 Patch Sets

This section describes the steps you must complete before starting the upgrade.

Shutting Down the Current Instance

Important: On multischema systems, first perform this task on the primary system, and then on the secondary systems.

To shut down BRM 7.5:
Creating a Complete Backup of Your BRM 7.5 Data

1. Stop all BRM 7.5 processes. Only the database instance should be running during the upgrade.

2. Ensure that no users are logged in.
   Users include customers, client applications, customer service representatives (CSRs), and so on.

Turning Off BRM Service Authentication and Authorization

**Important:** On multischema systems, first perform this task on the primary system, and then on the secondary systems.

To maintain a controlled environment for pre-upgrade testing, cut off interaction between your BRM system and your customers.

Creating a Complete Backup of Your BRM 7.5 Data

Creating a complete backup of the data in your BRM 7.5 environment requires the following actions to be performed in your BRM 7.5 environment:

1. **Backing Up BRM Files**
2. **Backing Up Your BRM 7.5 Database**

Backing Up BRM Files

Back up your BRM 7.5 files.

**Important:** If you are performing the upgrade on systems with distributed or HA architecture, back up the files on every node.

In particular, ensure that you back up the following files:

- All files customized for BRM 7.5 including any associated source code
- Registry files
- Policy files
- `pin.conf`
- `pin_setup.values`
- `Infranet.properties`
- All other files that contain any customizations used in BRM 7.5

**Important:** Back up all the customized load utility files to a different location. The data in these files is used to transfer and restore your customizations after upgrading to BRM 12.0.

For more information, see "Transferring Customizations to the New Release".
Backing Up Your BRM 7.5 Database

**Important:** On multischema systems, perform this task first on the primary database schema and then on the secondary database schemas.

Additionally, record the system time of each backup file so that you can match that time entry to import the backup data into the appropriate system in BRM 12.0.

Make a complete offline backup of your BRM database using the appropriate backup tools for your database version and ensure that the backup is completely valid and usable. The backup must contain both the database definition and all the database contents. See your database software documentation for more information on performing full database backups.

In addition to the backup, use the Oracle export `exp` utility to export all BRM 7.5 tables. This helps to restore individual tables, if necessary. For more information on the `exp` utility, see the discussion on export and import utilities in the appropriate version of the *Oracle Database Utilities* document.

**Important:** Store this backup in a safe location. The data in these files will become necessary if you encounter any issues in the upgrade process.

Creating Your BRM 12.0 Database

**Important:** On multischema systems, perform this task first on the primary database schema and then on the secondary database schemas.

Create a blank BRM database into which you will import the backup you created.

Upgrading to BRM 12.0 or 12.0 Patch Sets

This section describes the steps necessary to upgrade to BRM 12.0 or 12.0 Patch Sets.

Importing Your BRM 7.5 Data

After you have successfully created a new schema, create a BRM database user (which is same as the database user in BRM 7.5), and then import the backup data you created in "Backing Up Your BRM 7.5 Database".

**Important:** On multischema systems, perform this task first on the primary database schema and then on the secondary database schemas.

When you import the backup data (you created in "Backing Up Your BRM 7.5 Database"), check the system time of each backup file to ensure that the data is imported into the appropriate system in BRM 12.0.
Use the `imp` utility to import the backup data. For more information on the `imp` utility, see the discussion on export and import utilities in the appropriate version of the Oracle Database Utilities document.

### Installing Java

Install Java Platform, Standard Edition (Java SE), containing Java Development Kit (JDK) and Java Runtime Environment (JRE). The JRE is required for BRM installation. It is not included in the BRM 12.0 or 12.0 patch sets software pack.

See BRM Compatibility Matrix for the compatible version of Java.

For instructions on installing Java, see Oracle Java documentation.

### Installing Perl on Linux

Before installing Perl (32 bit) on Linux, do the following:

1. Install the required 32-bit libraries by doing the following:
   a. Go to the command prompt on the Linux server on which BRM 7.5 is installed.
   b. Run the following command:

   ```
   yum -y install libdb-devel
   yum -y install gdbm-devel
   ```

2. Set the environment variable for installing Perl by running the following command:

   - For Bash shell:

   ```
   export -n PATH
   export -n LD_LIBRARY_PATH
   export PATH=/usr/bin:/usr/local/bin:/bin
   ```

   - For C shell:

   ```
   unsetenv PATH
   unsetenv LD_LIBRARY_PATH
   setenv PATH /usr/bin:/usr/local/bin:/bin
   ```

To install Perl on Linux:

1. Download the source code for the compatible version of Perl to a temporary directory (`temp_dir`).

   See BRM Compatibility Matrix for the compatible version of Perl.

2. Go to the `temp_dir` directory and unzip the source code by running the following command:

   ```
   gunzip perl-version.tar.gz
   tar xf perl-version.tar
   ```

   where `version` is the compatible version of Perl.
3. Run the following command:

```bash
cd perl-version
sh Configure -des -Dcc="gcc -m32" -Dusethreads -Duserelocatableinc -Dprefix=
perl_path
```

where `perl_path` is the path to the directory in which you want to install Perl.

4. Run `make`.

5. Run `make test`.

6. Run `make install`.

7. Verify the Perl version by running the following command:

```bash
Perl -v
```

The Perl version is displayed.

If the latest version of Perl certified with BRM is not displayed, the latest Perl is not installed.

### Installing Perl on Solaris

Before installing Perl (32 bit) on Solaris, do the following:

1. Install Oracle Developer Studio. See `BRM Compatibility Matrix` for the compatible version and `Oracle Developer Studio Installation Guide` for installation instructions.

2. Comment the following entries in the `util.c` file:

```c
if(UNLIKELY(got != need)) {
    bad_handshake:"/* recycle branch and string from above */
    /* if(got != (void *)HSF_NOCHK)
       noperl_die("%s: loadable library and perl binaries are mismatched"
       " (got handshake key %p, needed %p)\n",
       file, got, need); */
}
```

3. Set the environment variable for installing Perl by running the following command:

   - For Bash shell:
     ```bash
     export -n PATH
     export -n LD_LIBRARY_PATH
     export PATH=/usr/bin:/usr/local/bin:/bin
     ```
   
   - For C shell:
     ```bash
     unsetenv PATH
     unsetenv LD_LIBRARY_PATH
     setenv PATH /usr/bin:/usr/local/bin:/bin
     ```

To install Perl on Solaris:

1. Download the source code for the compatible version of Perl to a temporary directory (`temp_dir`).

   See `BRM Compatibility Matrix` for the compatible version of Perl.

2. Go to the `temp_dir` directory and unzip the source code by running the following command:

```bash
gunzip perl-version.tar.gz
```
Upgrading BRM 12.0 or 12.0 Patch Sets

Upgrading BRM and Pipeline Manager

3. Run the following command:

```
tar xf perl-version.tar
```

where `version` is the compatible version of Perl.

3. Run the following command:

```
cd perl-version
sh Configure -des -Dcc="Oracle_dev_path" -Dusethreads -Dusemorebits -Dprefix=perl_path
```

where:
- `Oracle_dev_path` is the path to the directory in which the Oracle Developer Studio is installed.
- `perl_path` is the path to the directory in which you want to install Perl.

4. Run `make`.

5. Run `make test`.

6. Run `make install`.

7. Verify the Perl version by running the following command:

```
Perl -v
```

The Perl version is displayed.

If the latest version of Perl certified with BRM is not displayed, the latest Perl is not installed.

Obtaining ojdbc JAR File

Obtain the `ojdbc8.jar` jar file and save it in a directory on the machine on which BRM 7.5 is installed. Note the path to this directory; you are required to specify this path in the Specify Prerequisite Libraries Location screen during the BRM installation.

You can download this file from the "Oracle Database 12c Release 2 (12.2.0.1) JDBC Drivers & UCP Downloads" page on the Oracle Technology Network Web site:

```
```

Setting the Environment Variables

Before installing the upgrade, ensure that the environment variables are set correctly. To do so:

1. Set the JAVA_HOME environment variable to the directory in which the latest version of JRE certified with BRM is installed by running the following command:

```
setenv JAVA_HOME java_path
```

where `java_path` is the path to the directory in which the latest version of Java certified with BRM is installed; for example, `/Linux/x86_64/packages/jdk/jdk1.8.0_221`.

2. Verify the Java version by running the following command:

```
java -version
```

The Java version is displayed.
If the latest version of Java certified with BRM is not displayed, the latest JRE is not installed.

3. Set the PERL_HOME environment variable to the directory in which the latest version of Perl certified with BRM is installed by running the following command:

   setenv PERL_HOME Perl_path

   where Perl_path is the path to the directory in which the latest version of Perl certified with BRM is installed; for example, /perl_5_28_1/linux.

4. Set the PATH environment variable by running the following command:

   setenv PATH $JAVA_HOME/bin:$PERL_HOME/bin:${PATH}

5. Set the HOSTNAME environment variable by running the following command:

   setenv HOSTNAME hostName

   where hostName is the fully qualified host name of the machine on which BRM 7.5 is installed.

Setting BRM Wallet Location in sqlnet.ora

To set the BRM wallet location in the sqlnet.ora file:

1. Open the $ORACLE_HOME/network/admin/sqlnet.ora file on the machine on which BRM 7.5 is installed.

2. Add the following entries:

   SQLNET.WALLET_OVERRIDE = TRUE
   WALLET_LOCATION=(
       SOURCE=(METHOD=FILE)
       (METHOD_DATA=(DIRECTORY= $PIN_HOME/wallet/client))
   )

3. Save and close the file.

Obtaining the BRM Software

You can obtain and install the BRM 12.0 or BRM 12.0 patch sets complete software.

Obtaining the BRM 12.0 Software

To obtain the BRM 12.0 software:

1. Create a temporary directory (temp_dir).

2. Go to the Oracle software delivery Web site: http://edelivery.oracle.com

3. Sign in with your user name and password.

4. Search for Oracle Communications Billing and Revenue Management and select the BRM 12.0 software pack from the search results.

5. From the Platform list, select the platform, where platform is linux or solaris.

6. Download the BRM server pack to temp_dir.

7. Unzip the BRM server pack and extract the contents to temp_dir.
Obtaining the BRM 12.0 Patch Set Software
To obtain the BRM 12.0 Patch Set software:

1. Create a temporary directory (temp_dir).
2. Go to the My Oracle Support Web site:
   https://support.oracle.com
3. Sign in with your user name and password.
4. Click the Patches & Updates tab.
5. From the list, select Patch Name or Number.
6. In the text field, enter the PatchNumber and click Search, where PatchNumber is:
   - 28630668 for BRM 12.0 Patch Set 1
   - 30094488 for BRM 12.0 Patch Set 2
   The Patch Search Results page appears.
7. Click the patch name.
   The patch details appear.
8. From the Platform list, select the platform and click Download, where platform is linux or solaris.
   The File Download dialog box appears.
9. Download the appropriate pPatchNumber_12v_platform.zip software pack to temp_dir, where platform is linux or solaris.
   where v is the BRM patch set version. For example, 1 is the version for BRM 12.0 Patch Set 1 and 2 is the version for BRM 12.0 Patch Set 2.
10. Unzip the pPatchNumber_12v_platform.zip file.
    The zip file includes the brmserver_12.0.0.v0_platform_generic_full.jar file for installing the complete software.
    where v is the BRM patch set version. For example, 1 is the version for BRM 12.0 Patch Set 1 and 2 is the version for BRM 12.0 Patch Set 2.

Installing BRM and Pipeline Manager for Upgrade
To install BRM and Pipeline Manager for upgrade:

1. Obtain the BRM software. See "Obtaining the BRM Software".
2. Go to the temp_dir directory, and run one of the following commands:
   - To start the GUI installer:
     Java_home/bin/java -jar jar_file
     where:
     - Java_home is the directory in which you installed the latest compatible Java version.
     - jar_file is the BRM installer file. For example:
       For BRM 12.0, it is brmserver_12.0.0.0.0_platform_generic.jar.
       For BRM 12.0 patch sets, it is brmserver_12.0.0.v0_platform_generic_full.jar.
where:

\(v\) is the BRM patch set version. For example, 1 is the version for BRM 12.0 Patch Set 1 and 2 is the version for BRM 12.0 Patch Set 2.

*platform* is *linux* or *solaris*.

- To start the GUI installer and install BRM using the *oraInventory* directory in a different location:

  ```bash
  Java_home/bin/java -jar jar file -invPtrLoc FilePath/oraInst.loc
  ```

  where *FilePath* is the path to the directory in which the *oraInst.loc* file is located.

- To start the GUI installer and create installer logs for debugging:

  ```bash
  Java_home/bin/java -jar jar file -debug -logLevel INFO
  ```

- To start the GUI installer and create a silent installer response file during the installation:

  ```bash
  Java_home/bin/java -jar jar file -record -destinationFile path
  ```

  where *path* is the absolute path to the response file.

3. Click **Next**.

   The Installation Location screen appears.

4. Enter the full path or browse to the directory in which BRM 7.5 is installed.

5. Click **Next**.

   The Installation Type screen appears.

6. Select **Custom**, and click **Next**.

   The Feature Sets Selection screen appears.

7. From the components list, select **Upgrade Manager Framework** and then select the components that you already installed in BRM 7.5.

   **Upgrade Manager Framework** contains the BRM server upgrade and the Pipeline Manager upgrade.

   **Important:** Ensure that you select only the components in **Upgrade Manager Framework** and the components that you already installed in BRM 7.5 or BRM 7.5 patch sets. You can install any additional optional component after completing the BRM 12.0 upgrade. See "Adding Optional Components".

   If the existing BRM components are installed on separate machines, install the respective 12.0 component on the machine on which the existing component is installed. For example, if the BRM server and Pipeline Manager are installed on separate machines, install the BRM 12.0 server on the machine on which the existing BRM server is installed and BRM 12.0 Pipeline Manager on the machine on which existing Pipeline Manager is installed.

8. Click **Next**.
If you selected only the components in **Upgrade Manager Framework**, proceed to step 15.

If you selected the components in **Upgrade Manager Framework** and other components from the components list, the Specify Prerequisite Libraries Location screen appears.

9. Do the following:
   - In the **Prerequisite Libraries** field, enter the full path or browse to the directory in which the prerequisite libraries are stored.
   - Do one of the following:
     - If you do not want to enable secure communication between BRM server components, deselect the **Enable SSL for BRM server** check box.
     - To enable secure communication between BRM server components, leave the **Enable SSL** check box selected.

10. Click **Next**.
    The Oracle ZT PKI Encryption screen appears.

11. Do one of the following:
    - If you have enabled the Oracle ZT PKI Encryption enabled for BRM, select **Yes**.
    - If you have not enabled the Oracle ZT PKI Encryption for BRM, select **No** and proceed to step 14.

12. Click **Next**.
    The Specify BRM Wallet Details screen appears.

13. Enter the existing BRM wallet details:
   - In the **Wallet Password** field, enter the password for the BRM wallet.
   - In the **Wallet Location** field, enter the full path or browse to the directory in which the BRM wallet is located.

14. Click **Next**.

15. In the following screens, provide the requested information, and click **Next**.

For the description of the fields displayed, see "Installing All BRM Components" in **BRM Installation Guide**. Continue moving through the screens until the Installation Summary screen appears.

Your responses are written to the `BRM_home/setup/pin_setup.values` file, where `BRM_home` is the directory in which the BRM 12.0 server software is installed.
Upgrading Your Data to BRM 12.0

This section describes the tasks to be completed to upgrade your data to BRM 12.0:

Pointing the BRM 12.0 Installation to the New Database Schema

Before you run the BRM upgrade script, verify that the Oracle DM is configured correctly as this DM provides the interface to the BRM 12.0 database. Verify that Oracle DM is configured properly by starting `dm_oracle`.

Important: On multiscHEMA systems, first perform this task on the primary system and then on the secondary systems.

You should ensure that Oracle DM starts and stops correctly because upgrade scripts start and stop Oracle DM while making changes to the database schema.

Upgrading the BRM Database Schema to the BRM 12.0 Schema

Important: On multiscHEMA systems, first perform this task on the primary system and then on the secondary systems.
The following upgrade scripts run a series of scripts that upgrade BRM 7.5 to BRM 12.0 or 12.0 patch sets:

- `pin_upgrade_12.pl` upgrades to BRM 12.0.
- `pin_upgrade_12ps1.pl` upgrades to BRM 12.0 Patch Set 1.
- `pin_upgrade_12ps2.pl` upgrades to BRM 12.0 Patch Set 2.

To upgrade the BRM 7.5 (Portalbase) database schema to a BRM 12.0 database schema, complete the appropriate steps in this section.

**To upgrade the schema on single-schema systems:**

To upgrade the schema on single-schema systems:

1. Open the `BRM_home/setup/pin_tables.values` file in a text editor.
2. Set the values of the `PIN_CONF_PARTITION_IND` and `PIN_CONF_NON_EVENT_PARTITION_IND` parameters to "local".
   
   For example:
   ```
   # For indexes on event tables
   $PIN_CONF_PARTITION_IND = 'local';
   # For indexes on non event tables
   $PIN_CONF_NON_EVENT_PARTITION_IND = 'local';
   ```
3. Open the `BRM_home/setup/pin_setup.values` file in a text editor.
4. (Optional) Set the values of the following parameters:
   - `PIN_TEMP_DIR` to the directory in which you want to create the temporary files. Ensure that the directory is set with full write permissions.
   - `PIN_LOG_DIR` to the directory in which you want to create the BRM log files.
5. In the **Information about the databases** section, configure the database settings as required.
6. Set `SETUP_INIT_DB` to `YES`, if it is not set.
7. Save and close the file.
8. Run the `source` command on the `source.me` file:
   - For Bash Shell:
     ```
     % source source.me.sh
     ```
   - For C shell:
     ```
     % source source.me.csh
     ```
9. Go to the `BRM_home/setup/scripts` directory and run the following command:

    `perl pin_upgrade_12.pl`

    For upgrading to BRM 12.0 Patch Set 1:
    ```
    % perl pin_upgrade_12ps1.pl
    ```

    **Important:** Ensure that you do not run the upgrade script in the background.
% perl pin_upgrade_12ps1.pl

For upgrading to BRM 12.0 Patch Set 2:
% perl pin_upgrade_12ps2.pl

10. At the command prompt, enter the BRM wallet password.

11. Merge the contents of the backed up pin_ctl.conf file into the new pin_ctl.conf file.

Upgrading the Schema on Multischema Systems

To upgrade the schema on multischema systems, you upgrade the primary database schema and then the secondary database schemas.

To upgrade the primary database schema, complete the following on your primary BRM installation machine:

1. Open the BRM_home/setup/pin_tables.values file in a text editor.

2. Set the values of the PIN_CONF_PARTITION_IND and PIN_CONF_NON_EVENT_PARTITION_IND parameters to "local".
   For example:
   # For indexes on event tables
   $PIN_CONF_PARTITION_IND = "local";
   # For indexes on non event tables
   $PIN_CONF_NON_EVENT_PARTITION_IND = "local";

3. Open the BRM_home/setup/pin_setup.values file in a text editor.

4. (Optional). Set the values of the following two parameters.
   ■ Set PIN_TEMP_DIR to the directory in which you want to create the temporary files.
   ■ Set PIN_LOG_DIR to the directory in which you want to create the BRM log files.

5. In the Information about the databases section, configure the database settings as required for your primary database schema.

6. Set SETUP_INIT_DB to YES, if it is not set.

7. Save and close the file.

8. Go to BRM_home and source the source.me file:
   ■ For Bash Shell:
     % source source.me.sh
   ■ For C shell:
     % source source.me.csh

9. Go to the BRM_home/apps/multi_db directory and run the following command:
   ./install.sh

10. Follow the on-screen instructions and enter the following information for the primary schema and for each secondary schema when requested:
   ■ Schema user name
- Schema password
- Schema SID (the BRM database alias of the schema)

**Important:** Repeat the "Do you have secondary schema to process" step for each secondary schema in your system.

The install script fixes any data errors caused by conflicting storable class IDs. The errors might have occurred during global (XA) transactions involving multiple schemas or when accounts were migrated from one schema to another.

**Note:** As the install script runs, it generates the `BRM_home/apps/multi_db/fix_multi_schema.log` file. To view the progress of the script, display the log file in a different console window.

11. Go to the `BRM_home/setup/scripts` directory and run the following command:
   For upgrading to BRM 12.0:
   ```
   % perl pin_upgrade_12.pl
   ```
   For upgrading to BRM 12.0 Patch Set 1:
   ```
   % perl pin_upgrade_12ps1.pl
   ```
   For upgrading to BRM 12.0 Patch Set 2:
   ```
   % perl pin_upgrade_12ps2.pl
   ```

12. At the command prompt, enter the BRM wallet password.

13. Merge the contents of the backed up `pin_ctl.conf` file into the new `pin_ctl.conf` file.

14. Generate the root key by running the following command:
    ```
    pin_crypt_app -useZT -genkey
    ```
    The root key is generated.

15. Set the generated root key in the Oracle DM configuration file (`BRM_home/sys/dm_oracle/pin.conf`).

To upgrade each secondary database schema, perform the following on each secondary BRM installation machine:

1. Before you install the secondary schema, set the BRM_WALLET environment variable to the BRM wallet in the primary schema:
   ```
   setenv BRM_WALLET BRM_home/wallet/client
   ```

2. Open the `BRM_home/setup/pin_setup.values` file in a text editor.

3. Change the following entry:
   ```
   $DM_ORACLE(‘db_num’)
   ```
   where `db_num` is the number of the corresponding secondary database schema to upgrade.
4. In the **Information about the databases** section, configure the database settings as required.

5. Set **SETUP_INIT_DB** to **YES**, if it is not set.

6. Save and close the file.

7. Go to **BRM_home** and source the **source.me** file:
   - For Bash Shell:
     
     % source source.me.sh
   - For C Shell:
     
     % source source.me.csh

8. Go to the **BRM_home/setup/scripts** directory and run the following command:
   - For upgrading to BRM 12.0:
     
     % perl pin_upgrade_12.pl
   - For upgrading to BRM 12.0 Patch Set 1:
     
     % perl pin_upgrade_12ps1.pl
   - For upgrading to BRM 12.0 Patch Set 2:
     
     % perl pin_upgrade_12ps2.pl

9. At the command prompt, enter the BRM wallet password.

10. Merge the contents of the backed up **pin_ctl.conf** file into the new **pin_ctl.conf** file.

11. Set the root key of the primary schema in the Oracle DM configuration file (**BRM_home/sys/dm_oracle/pin.conf**).

To reconfigure your entire multischema system, perform the following on the primary BRM installation machine:

1. Go to the **BRM_home/setup/scripts** directory and run the following command:

   pin_multidb.pl -i

2. At the following prompt, enter **y**, which begins the configuration:

   Do you want to start the configuration now? (y/n):  y

3. At the following prompt, enter **2**, which initializes the primary database:

   Please enter the starting step (0-8). If you don't know, enter 0:  2

4. Follow the instructions displayed and exit the **pin_multidb.pl** script.

   At the prompt to enter the wallet password, enter the wallet password for the primary schema.

   The refresh groups are recreated in the primary database.

5. Run the following command:

   pin_multidb.pl -R all

   The views, based on the information in the refresh groups, are re-created in the secondary databases.
Upgrading the Pipeline Manager Database Schema to the BRM 12.0 Schema

To upgrade the Pipeline Manager database schema:

---

**Important:** Run the database upgrade script on the primary schema and then on the secondary schemas.

---

1. Open the `Pipeline_home/upgrade/pipeline_upgrade.cfg` file in a text editor, where `Pipeline_home` is the directory in which Pipeline Manager is installed.

2. Set the values of the following parameters:
   - Set `PIN_TEMP_DIR` to the directory in which you want to create the temporary files.
   - Set the `$PIPELINE_TBLSPACE` environment variable to the tablespace where you want to create pipeline database objects.

3. In the **Information about the databases** section, configure the following database settings as required:

   ```
   $MAIN_DB{'vendor'} = "oracle";
   $MAIN_DB{'alias'} = ( $ENV{'ORACLE_SID_PIN'} or $ENV{'ORACLE_SID'} ) ;
   $MAIN_DB{'user'} = "USERNAME";
   $MAIN_DB{'Database'} = "DATABASE_NAME";
   $MAIN_DB{'Host'} = ( $ENV{'ORACLE_SID_PIN'} or $ENV{'ORACLE_SID'} ) ;
   ```

4. Save and close the file.

5. Grant the required access to user `pin` on the Pipeline Manager tables and sequences, if you have not already done so. See the discussion about loading the discount stored procedure in BRM Installation Guide for more information about the Pipeline Manager tables and sequences that you should grant user `pin` access to.

6. Go to the `Pipeline_home/setup/scripts` directory and run the following command:

   **Important:** Ensure that you do not run the upgrade script in the background.

   
   For upgrading to BRM 12.0:
   ```
   % perl pin_upgrade_12.pl
   ```
   
   For upgrading to BRM 12.0 Patch Set 1:
   ```
   % perl pin_upgrade_12ps1.pl
   ```
   
   For upgrading to BRM 12.0 Patch Set 2:
   ```
   % perl pin_upgrade_12ps2.pl
   ```

7. Go to the `Pipeline_home/database/Oracle/Scripts` directory and run the following command:

   ```
   sqlplus login/password@database_alias < create_pricing_discountmodel_procedures.plb
   ```

   where:
Completing the Upgrade to BRM 12.0 or 12.0 Patch Sets

This section describes the tasks required to complete the upgrade process.

Installing the BRM 12.0 Client Applications

When you install client applications, be sure to update the BRM 12.0 Infranet.properties and INI files with any 12.0 customizations. See "Installing BRM Clients" in BRM Installation Guide for instructions.

---

**Important:** You must upgrade all existing client applications and optional components to BRM 12.0 or a BRM 12.0 patch set before you install additional client applications and optional components in BRM 12.0 or a BRM 12.0 patch set.

---

To upgrade custom client applications, recompile them with BRM 12.0 libraries. See "Updating Custom Applications" for more information.

Adding Optional Components

At this point, add the optional components that you plan to use in BRM 12.0 or 12.0 patch sets. See “Installing Individual BRM Components” in BRM Installation Guide for installing new optional components in BRM 12.0 or 12.0 patch sets.

---

**Important:** Before you install a new optional component, in case you changed the database password after the previous installation, ensure that the database password is the same in all the configuration entries in the Oracle wallet. To change the database password in the configuration entries, see "About Oracle Wallet" in BRM System Administrator’s Guide for more information.

---

Updating Configurations

Complete the steps in this section that are appropriate for your BRM 12.0 environment.

**Update DM_IFW_SYNC Queue Database Configuration**

If you use Pipeline Manager in BRM 12.0, update the DM_IFW_SYNC queue database configuration.

To update the DM_IFW_SYNC queue database configuration:

1. Open the BRM_home/sys/dm_ifw_sync/pin.conf file in a text editor.
2. Set the following values:
   
   ```
   dm_ifw_sync sm_database DATABASE_NAME or DATABASE_SID
   dm_ifw_sync sm_id USERNAME
   ```
3. Save and close the file.
Verify IFW_SYNC_QUEUE Queue in the Database
To verify that the IFW_SYNC_QUEUE queue is created in the required (BRM or Pipeline) database:

1. Go to the required directory by running the following command:
   \[ \text{cd } \text{PIN_HOME/sy/dm_ifw_sync/} \]

2. Run the following command:
   \[ \text{pin_ifw_sync_oracle.pl create -q queue_name -t queue_table -l login/password@ORACLE_SID} \]

   where:
   - \text{queue_name} is the queue name for the \text{-q} parameter.
   - \text{queue_table} is the table name for the \text{-t} parameter.
   - \text{login} is the username for the Portalbase database.
   - \text{password} is the Portalbase database user password.
   - \text{ORACLE_SID} is the Portalbase database alias.

Update the Default Settings for the pin_rel Utility
If you use the \text{pin_rel} utility, ensure that the database configurations settings are updated.

To update the default settings for the \text{pin_rel} utility:

1. Go to the \text{BRM_home/apps/pin_rel} directory.
2. Open the \text{Infranet.properties} file.
3. Check and, if necessary, update the entries for the following:
   \[
   \text{infranet.rel.dbname = DATABASE_NAME} \\
   \text{infranet.rel.userid = USERNAME} \\
   \]
4. Save and close the file.

Updating the Settings for the pin_virtual_time Utility
If you are testing the upgrade on a test system, ensure that you uncomment all entries for the \text{pin_virtual_time} utility in the configuration files associated with your BRM test system.

If you are upgrading your production system, ensure that you remove all entries for the \text{pin_virtual_time} utility from the configuration files associated with your BRM production system.

Upgrading AMM in a Multischema System
If you have a multischema system and it includes Account Migration Manager (AMM), you must run the AMM installation scripts to upgrade your multischema system.

To upgrade AMM in a multischema system:

1. Log in as user \text{pin} and go to the \text{BRM_home/setup/scripts} directory.
2. Run the \text{pin_amt_install} script:
   \[
   \% \text{perl pin_amt_install.pl} \\
   \]
Post-Upgrade Procedures

Restoring Customizations

**Important:** You must first incorporate customizations on the primary schema and then on the secondary schemas.

Incorporate any customizations you made to your Release 7.5 policy source code, configuration files, invoicing, reports, and general ledger reporting. See "Transferring Customizations to the New Release" for more information.

Post-Upgrade Procedures

This section provides a list of post-upgrade tasks. Some tasks are optional or apply only to certain platforms or system configurations. Be sure to check whether a task is required for your system.

Perform these tasks after you upgrade your BRM system:

1. **Creating an Oracle AQ Database Queue**
2. **Loading the pin_notify File**
3. **Copying Files for Compiling CMs**
4. **Setting the Environment Variables for Pipeline Manager**

Additionally, perform these tasks if you are upgrading directly to BRM 12.0 patch sets:

- Configuring SSL for the BRM Database
- Running create_procedures Script
- Merging the Price Lists
- Setting the DM_IFW_SYNC Queue Database and CM Configuration Entries

Creating an Oracle AQ Database Queue

**Important:** On multischema systems, first perform this task on the primary system and then on the secondary systems.

If your system includes the Synchronization Queue Data Manager, you must re-create your Oracle AQ database queue before you can begin synchronizing pricing data. For more information on how to create a database queue, see “Manually Creating a Database Queue on Oracle” in *BRM Synchronization Queue Manager*.

After you re-create the Oracle AQ database queue, perform these steps:

1. Update `BRM_home/sys/dm_aq/aq_queue_names` with the re-created Oracle AQ database queue name.
2. Restart DM_AQ services.

Loading the pin_notify File

**Important:** On multischema systems, first perform this task on the primary system and then on the secondary systems.
If your system includes Pipeline Manager, you must reload the event notification configuration file, **pin_notify**, in the BRM database. For more information, see “About the Event Notification List” in BRM Developer’s Guide.

To load the **pin_notify** file:

1. Go to the `BRM_home/sys/data/config` directory.
2. Use the following command to run the `load_pin_notify` utility:
   ```
   load_pin_notify pin_notify
   ```
   If you do not run the utility from the directory in which the configuration file is located, include the complete path to the file.
3. Stop and restart the Connection Manager (CM).

### Copying Files for Compiling CMs

You must copy some files required for building versions of the CM.

To copy the required files for compiling CMs:

1. Copy the following files from the `BRM_home/lib` directory to the `BRM_Home/PortalDevKit/source/sys/cm` directory:
   - `libcmpin.so`
   - `libdmpin.so`
2. Copy the `cm.cpp` file from the `BRM_home/source/sys/cm` directory to the `BRM_Home/PortalDevKit/source/sys/cm` directory.
3. Open the `BRM_Home/PortalDevKit/source/samples/env.unix` file in a text editor.
4. Add or modify the following environment variables:
   ```
   LIBDIR = BRM_home/PortalDevKit/lib
   RW_INCDIR = PortalDevKit_hostname/rwWorkspace
   INCDIR = BRM_home/PortalDevKit/include
   PCM_JAR = BRM_home/jars/pcm.jar
   JDK_HOME = jdk_path
   PCMEXT_JAR = BRM_home/jars/pcmext.jar
   ```
   where:
   - `hostname` is the name of the machine on which the BRM server is installed.
   - `jdk_path` is the path to the directory in which the latest version of JRE certified with BRM is installed; for example, `/Linux/x86_64/packages/jdk/jdk1.8.0_144`.
5. Save and close the file.

### Setting the Environment Variables for Pipeline Manager

If your system includes Pipeline Manager and if you have the Oracle ZT PKI Encryption enabled in BRM 7.5, you must set some environment variables for Pipeline Manager.

On the machine on which you installed the BRM 12.0 server, set the `BRM_WALLET` environment variable for pipelines by running the following command:

```
setenv BRM_WALLET Pipeline_home/wallet/client
```
Configuring SSL for the BRM Database

**Important:** SSL-enabled database is not supported on multischema systems.

On single-schema systems, you can configure secure sockets layer (SSL) for the BRM database by creating wallets for storing certificates and then modifying the following configuration files in the Oracle_home/network/admin directory, where Oracle_home is the directory in which the Oracle database is installed, to point to the appropriate wallet:

- sqlnet.ora
- tnsnames.ora
- listener.ora

You can use the Orapki utility to create the wallets.

For information about configuring SSL for the Oracle database, see the Oracle Database documentation.

Running create_procedures Script

After installing the BRM patch set, you must run the `create_procedures` script to remove any invalid objects in the BRM database.

In a multischema system, perform this task on the primary Oracle DM machine.

To run the `create_procedures` script:

1. Go to the `BRM_home/sys/dm_oracle/data` directory.
2. Run the following command, which opens SQL*Plus:

   ```
   sqlplus login/password@ORACLE_SID
   ```

   where:
   - `login` is the user name for the BRM database schema.
   - `password` is the password for the specified user name.
   - `ORACLE_SID` is the database alias of the BRM database schema.

3. Run the following command:

   ```
   @create_procedures_character_set.plb
   ```

   where `character_set` specifies the database character set of either UTF8 or AL32UTF8.

Merging the Price Lists

You must perform this task only if you were using branding in BRM 7.5.

Branding is not supported in BRM 12.0. Therefore, if you have upgraded BRM 7.5 to BRM 12.0 or if you are upgrading BRM 7.5 to the BRM 12.0 patch set, you must merge the brand price lists.

To merge the price lists:

1. Go to the `BRM_home/sys/dd/data` directory.
2. Run the following command, which opens SQL*Plus:

```
sqlplus login/password@ORACLE_SID
```

where:
- `login` is the user name for the BRM database schema.
- `password` is the password for the specified user name.
- `ORACLE_SID` is the database alias of the BRM database schema.

3. Run the following command:

```
@merge_price_list.sql
```

### Setting the DM_IFW_SYNC Queue Database and CM Configuration Entries

You must add the required DM_IFW_SYNC Queue Database and CM configuration entries to the corresponding `pin.conf` files.

To set the DM_IFW_SYNC queue database and CM configuration entries:

1. Open the `BRM_home/sys/dm_ifw_sync/pin.conf` file.
2. Add the following required configuration entries:

```
- dm pcm_connect_max_retries 1
- dm pcm_reconnect_max_retries 3
- dm pcm_op_max_retries 0
- dm pcm_op_timeout_max_retries 0
- dm pcm_timeout_in_msecs -1
- dm pcm_auto_reconnect 1
- dm pcm_suspect_timeout_in_msecs -1
- dm pcm_bad_connection_retry_delay_time_in_secs 0
- dm pcm_reconnect first
- -- custom_field_caching 0
  -- ops_fields_extension_file dummy
```

3. Save and close the file.
4. Open the `BRM_home/sys/cm/pin.conf` file.
5. Add the following required configuration entries:

```
- cm pcm_connect_max_retries 1
- cm pcm_reconnect_max_retries 3
- cm pcm_op_max_retries 0
- cm pcm_op_timeout_max_retries 0
- cm pcm_reconnect first
- cm pcm_timeout_in_msecs -1
- cm pcm_suspect_timeout_in_msecs -1
- cm pcm_reconnect first
- cm pcm_bad_connection_retry_delay_time_in_secs 0
- cm pcm_connect_rcvbuf_size 87380
- cm keepalive 1
- cm log_session 0
```

6. Save and close the file.
Restoring Service to Your Customers

The Oracle Communications Online Charging Solution document contains the details on the steps that must be completed to restore service to your customers. Complete the process that is appropriate for your BRM 12.0 installation.
Part IV
Migrating Pricing Data from The BRM Database to PDC

This part includes information about migrating pricing data from the Oracle Communications Billing and Revenue Management (BRM) database to PDC.

Part I contains the following chapters:

- Migrating Pricing Data to PDC With ECE
- Migrating Pricing Data to PDC With RRE and BRE
- Migration Reports
- BRM Pricing Migration Utility
Migrating Pricing Data to PDC With ECE

This chapter describes how to set up your environments and run the Oracle Communications Pricing Design Center (PDC) migration utility to migrate Oracle Communications Billing and Revenue Management (BRM) pricing data or PDC pricing data to a PDC system that supports BRM Elastic Charging Engine (ECE).

- If you are migrating BRM pricing data to PDC with ECE, see "About Migrating BRM Pricing Data to PDC With ECE".
- If you are migrating PDC pricing data to PDC with ECE, see "About Migrating PDC Pricing Data to PDC With ECE".

Pricing data migration should be performed by experienced pricing administrators. You should have a basic understanding of the following:

- Creating product offerings. See the discussion about creating product offerings in PDC Creating Product Offerings.
- Configuring setup components. See the discussion about configuring setup components in PDC Creating Product Offerings.

To use the migration utility, you must also understand the following:

- Database administration tasks
- Extensible Markup Language (XML) programming
- XML schema definition (XSD)

To minimize impact and risks to your production systems, Oracle recommends to perform the initial migration on a development test system and then import the migrated data to your production systems.

About Migrating BRM Pricing Data to PDC With ECE

If you are using Pricing Center to create and manage your price lists and using the real-time engine (RRE) and the batch rating engine (BRE) for usage rating, the pricing data is stored in the BRM database. If you want to use ECE for usage rating, you must first install PDC with the ECE option and then migrate the pricing data from the BRM database into the PDC database.

You use the command-line utility, MigrateBRMPricing, to migrate the BRM pricing data into the PDC database.

**Note:** The migration of branded data is not supported.
The migration utility migrates both real-time and batch pricing data. The BRM pricing data consists of configuration objects and pricing objects. Because the pricing objects reference configuration objects, you must migrate the configuration objects prior to migrating the pricing objects. To do this, you run the migration twice: first to migrate the configuration objects and then to migrate the pricing objects.

The configuration objects include data such as event objects, service objects, impact categories, and zone models.

The pricing objects consist of the following:

- Products and all objects referenced by a product, including rateplans, price model selectors, Access Point Name (APN) selectors, price models, time models, usage scenario (USC) groups.
- Discounts and all objects referenced by discounts, including discount models, discount model selectors, discount rules, triggers, and filters.
- Chargeshares and all objects referenced by a chargeshare.
- Subscriptions objects, including deals, plans, and plan lists.

If the pricing analysis reports contain a large number of messages, you can consider migrating pricing objects by category. If you decide to migrate the pricing objects by category, you must migrate them in the following order:

1. Products and all objects referenced in the products
2. Discounts and all the objects referenced in the discounts
3. Chargeshares
4. Subscription objects

If you migrate pricing objects by category, each migration process must be completed first before starting the next. For example, the products migration process must be completed, that means all products and related objects are analyzed, transformed, and available in the PDC database, before you start the discounts migration process and so on.

See "MigrateBRMPricing" for the utility’s syntax and parameters.

---

**Note:** The BRM term for an object sometimes differs from the PDC term for the same object. See the mapping BRM to PDC terminology table in *PDC Creating Product Offerings* for more information.

---

Migration of the BRM pricing data is performed in two phases:

1. **Data analysis:** In this phase, the migration utility extracts and analyzes the BRM pricing data for PDC compatibility. The analysis shows errors that need to be fixed before the data can be migrated.

2. **Data migration:** In this phase, the migration utility adjusts, transforms, checks for ECE compatibility, adjusts (if needed), and migrates the BRM pricing data to the PDC database.

You must perform the migration process twice. First you analyze and migrate configuration objects, such as event objects, service objects, impact categories, and zone models. Then, you analyze and migrate pricing objects, such as products and discounts.
About the BRM Data Analysis Phase for Migrating BRM Pricing Data to PDC With ECE

The data analysis phase is the first phase of the migration of the BRM pricing data. The following procedure describes how the BRM pricing data is processed in this phase:

1. The migration utility extracts the BRM pricing data from the BRM database and saves the data in XML files.
2. The migration utility analyzes the data to determine if the configurations are supported in PDC and whether any configurations need to be adjusted prior to the migration.
3. The migration utility generates analysis reports that provide the list of the data configurations that need to be adjusted by the utility during the migration process and any configurations that you would need to adjust manually.

You need to review the analysis reports and resolve any pricing data configuration errors in the BRM database. If you make updates to the pricing data in the BRM database, you must restart the migration process to extract the latest objects from the BRM database and perform the data analysis again.

When you restart the migration process, the analysis reports are also re-generated for the latest data.

Figure 9–1 shows the data analysis phase for migrating BRM pricing data to PDC with ECE:

![Data Analysis Phase for Migrating BRM Pricing Data to PDC With ECE](image)

The migration utility can adjust most BRM pricing configurations. Adjustments to the BRM pricing data configurations may result in existing BRM objects being updated or new BRM objects being created. In some cases, the migration utility may require additional information from you before it can adjust the data. But, there are some configurations that it cannot adjust and for those cases, you may need to change the configurations manually before continuing with the migration.

The data analysis phase is complete when there are no data configurations errors in the analysis reports.

About the Data Migration Phase for Migrating BRM Pricing Data to PDC With ECE

The data migration phase is the second phase of the migration process. You must resume the migration process after the analysis phase is complete to start the data migration.

The following procedure describes how the data is processed in the data migration phase:

1. The migration utility adjusts the BRM pricing data in the XML files, if necessary, to enable transformation to the PDC data model.
2. The migration utility transforms the BRM pricing data to PDC components and saves them in XML files.

3. The migration utility checks the data to determine if the configurations are supported in ECE and whether any configurations need to be adjusted prior to the migration.

4. The migration utility generates compatibility reports that provide the list of the data configurations that need to be adjusted by the utility during the migration process and any configurations that you would need to adjust manually.

You need to review the compatibility reports, resolve any pricing data compatibility errors in the BRM database, and restart the data migration process.

When you restart the migration process, the compatibility reports are also re-generated for the latest data.

5. The migration utility loads the PDC components into the PDC database.

6. The migration utility loads any new and updated BRM objects (resulting from the data adjustments) into the BRM database.

7. The migration utility generates the migration report with information about the BRM objects that were successfully migrated to the PDC database.

8. The migration utility generates the process report with information about the BRM objects that were created or updated in the BRM database.

For more information about the migration utility reports, see "About Migration Reports".

Figure 9–2 shows the data migration phase for migrating BRM data to PDC with ECE:

![Data Migration Phase for Migrating BRM Data to PDC With ECE](image)
For more information on setting up and migrating the BRM pricing data to PDC with ECE, see "Performing Migration of BRM Pricing Data to PDC With ECE".

Performing Migration of BRM Pricing Data to PDC With ECE

To set up and migrate the BRM pricing data to PDC with ECE:

1. Create a backup of the BRM production database.
   See "Creating a BackUp of Your BRM Database".
2. Setup the development system to run the initial migration of the BRM pricing data.
   See "Setting Up the BRM Development System" for more information.
3. Migrate the BRM configuration and pricing objects.
   See “Migrating the BRM Pricing Data to PDC With ECE” for more information.
4. Restart the PDC and ECE systems.
   See "Restarting the Systems".
5. Test the migrated data.
   See "Testing the Migrated Data".
6. Import the migrated data to the production systems.
   See "Importing the Migrated Data to the Production Systems".

Creating a BackUp of Your BRM Database

If you migrating data from the BRM database, Oracle recommends that you perform a full backup of your BRM production database prior to running the migration.

If you migrating data from the PDC database, Oracle recommends that you perform a full backup of your PDC production database prior to running the migration.

Performing a full backup allows you to restore your production database to its original state in the event of any data corruption or data loss. Back up both the database definition and all the database contents.

Use the backup to restore the database on a test system to ensure the backup is valid.

See your database software documentation for more information on performing full database backups.

Setting Up the BRM Development System

Setting up the BRM development system includes setting up a BRM system with BRM pricing data and installing a PDC system.

Perform the following tasks to setup your development system:

1. Review the system requirements for installing the PDC and BRM systems.
   See the discussion about Pricing Design Center system requirements in PDC Installation and System Administration Guide.
2. Perform the following pre-installation tasks for PDC:
   ■ Installing and configuring the Oracle Database
   ■ Installing and configuring Oracle WebLogic Server
Performing Migration of BRM Pricing Data to PDC With ECE

- Installing and configuring BRM
  See the discussion about Pricing Design Center pre-installation tasks in *PDC Installation and System Administration Guide*.

3. Install the complete PDC software.

**Important:** Before you install PDC, ensure that the PDC database does not have any existing pricing data.

When you install BRM Integration Pack, ensure that you select the Migrate RRE and BRE pricing data to PDC with ECE option.

See the discussion about installing Pricing Design Center complete software in *PDC Installation and System Administration Guide* for more information.

4. Copy the BRM pricing data from the production database to the development system.

You can copy the BRM pricing data in one of two ways:

- Use the BRM load utilities, `loadpricelist` and `LoadIfwConfig`, to export the pricing data from your BRM production database and load it into the BRM development database.
  
  See the discussion about pricing utilities in *PDC Creating Product Offerings* for information about running these utilities.

**Note:** Keep a record of the following information as you create the BRM database. This information is used later to configure the MigrateBRMPricing utility.

- User login and password for the BRM database
- Name of the machine on which BRM database is created
- IP address of the machine on which BRM database is created
- Port number assigned to the BRM database
- Service Name or SID for the BRM database

5. Ensure `BRM_Integration_Pack_Home/apps/bin` is in your PATH environment variable, where `BRM_Integration_Pack_Home` is the directory in which you installed BRM Integration Pack.

6. (Optional) Configure the MigrateBRMPricing utility. See "Configuring the Migration Utility" for more information.

7. Ensure all users of the migration utility are added to the Migration Admin group.

The users of the migration utility must belong to the Migration Admin group, which is created during the PDC installation. You can add additional users to this group by using the WebLogic Administration Console.

8. Ensure that the BRM `loadpricelist` and `LoadIfwConfig` utilities are configured correctly and can connect to the BRM database.
The migration utility uses the BRM load utilities to extract the pricing data from the BRM database to XML files.

See the discussion about pricing utilities in *PDC Creating Product Offerings* for more information.

9. (Optional) If your BRM data consists of pipeline services that are mapped to multiple events in the Pipeline Manager IFW_REF_MAP database table, then you must add REF_PARAM combination key to the IFW_REF_MAP entry in *IFW_Home/tools/XmlLoader/LoadIfwConfig.xsd* file.

```xml
<xs:element name="IFW_REF_MAP" type="TableType_IFW_REF_MAP" minOccurs="0" maxOccurs="unbounded">
  <xs:key name="IFW_REF_MAP_PrimaryKey_1">
    <xs:selector xpath="IFW_REF_MAP'/>
    <xs:field xpath="@ID'/>
    <xs:field xpath="@REF_OBJ'/>
    <xs:field xpath="@REF_PARAM'/>
  </xs:key>
</xs:element>
```

10. Obtain the Java Keystore password.

**Configuring the Migration Utility**

The migration configuration file contains the information that you provided during PDC installation. You can edit this file if you want to change the information that you provided.

To configure the migration utility:

1. Make a copy of the *BRM_Integration_Pack_Home/apps/migration/MigrateConfiguration.xml* file, where *BRM_Integration_Pack_Home* is the directory in which you installed BRM Integration Pack.

2. Open the copy in a text editor.

3. Edit the file based on your requirements.

   Table 9–1 lists the elements in *MigrateConfiguration.xml* and the syntax and description for each element.
### Table 9–1  Elements in the MigrateConfiguration.xml File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>walletConfiguration</td>
<td><code>&lt;walletConfiguration&gt;</code>&lt;pdcWalletLocation&gt;BRMIntegrationPackWallet&lt;/pdcWalletLocation&gt;<code>walletConfiguration</code></td>
<td>Specifies the Oracle wallet that stores the sensitive information, such as database passwords, where <code>BRMIntegrationPackWallet</code> is the complete path to the BRM Integration Pack wallet.</td>
</tr>
</tbody>
</table>
| xrefDatabase    | `<xrefDatabase>`<connectionInfo><login>CrossRefUserName</login><hostName>CrossRefHostName</hostName><port>CrossRefPort</port><serviceName>CrossRefServiceName</serviceName></connectionInfo>`xrefDatabase`  | Contains the details about the transformation cross-reference database, where:  
- `CrossRefUserName` specifies the cross-reference database user name  
- `CrossRefHostName` specifies the IP address or the host name of the machine on which the cross-reference database is configured  
- `CrossRefPort` specifies the port number assigned to the cross-reference database  
- `CrossRefServiceName` specifies the name of the cross-reference database service |
| migrationDatabase | `<migrationDatabase>`<connectionInfo><login>MigrationUserName</login><hostName>MigrationHostName</hostName><port>MigrationPort</port><serviceName>MigrationServiceName</serviceName></connectionInfo>`migrationDatabase`  | Contains the details about the migration cross-reference database, where:  
- `MigrationUserName` specifies the cross-reference database user name  
- `MigrationHostName` specifies the IP address or the host name of the machine on which the cross-reference database is configured  
- `MigrationPort` specifies the port number assigned to the cross-reference database  
- `MigrationServiceName` specifies the name of the cross-reference database service  

By default, the connection information in the `<migrationDatabase>` elements is same as the connection information in the `<xrefDatabase>` elements. However, during BRM Integration Pack installation, the Installer allows you specify a migration cross-reference database that is different from the transformation cross-reference database. In that case, the connection information in the `<migrationDatabase>` elements and the `<xrefDatabase>` elements may differ. |
| migrationSource | `<migrationSource>`<deploymentType>MigrationSourceType</deploymentType><pricingServer>SourcePDCPricingServer</pricingServer><brmConfiguration>SourceBRMConfiguration</brmConfiguration>`migrationSource`  | Contains the details about the migration source system.  
- `<deploymentType>` element specifies the migration source, where `MigrationSourceType` is:  
  - `LEGACY_BRM`, to specify that the source is a BRM system that uses real-time and batch rating engines and BRM Pricing Center.  
  - `PDC_WITH_ONLINE_BATCH_USAGE`, to specify that the source is a BRM system that uses real-time and batch rating engines and PDC.  
- `<pricingServer>` element contains the source PDC server information if the migration source is `PDC_WITH_ONLINE_BATCH_USAGE`.  
- `<brmConfiguration>` contains the source BRM system configuration details. |
Table 9–1 (Cont.) Elements in the MigrateConfiguration.xml File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pricingServer</td>
<td>&lt;migrationSource&gt;</td>
<td>Contains the source PDC server information, where:</td>
</tr>
<tr>
<td></td>
<td>&lt;pricingServer&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;connectionInfo&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;hostName&gt;SourcePricingServerHostName&lt;/hostName&gt;</td>
<td>- <strong>SourcePricingServerHostName</strong> specifies the IP address or the host name of the machine on which the source PDC server is deployed.</td>
</tr>
<tr>
<td></td>
<td>&lt;port&gt;SourcePricingServerPort&lt;/port&gt;</td>
<td>- <strong>SourcePricingServerPort</strong> specifies the port number of the domain on which the source PDC server is deployed.</td>
</tr>
<tr>
<td></td>
<td>&lt;adminUser&gt;SourceAdminUserName&lt;/adminUser&gt;</td>
<td>- <strong>SourceAdminUserName</strong> specifies the user name of the source PDC server administrator.</td>
</tr>
<tr>
<td></td>
<td>&lt;pdcUser&gt;SourcePDCUser&lt;/pdcUser&gt;</td>
<td>- <strong>SourcePDCUser</strong> specifies the user name of the source PDC system user.</td>
</tr>
<tr>
<td></td>
<td>&lt;pdcSSL&gt;SourceSSLOption&lt;/pdcSSL&gt;</td>
<td>- <strong>SourceSSLOption</strong> specifies whether the source PDC server supports SSL.</td>
</tr>
<tr>
<td></td>
<td>&lt;/connectionInfo&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;/pricingServer&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;/migrationSource&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**enabled.** Specifies that PDC supports SSL. If SSL is enabled, **MigrateBRMPricing** uses the `t3s://Host:Port` URL to access PDC.

**disabled.** Specifies that PDC does not support SSL. If SSL is disabled, **MigrateBRMPricing** uses the `t3://Host:Port` URL to access PDC.
### Table 9–1 (Cont.) Elements in the MigrateConfiguration.xml File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>brmConfiguration</code></td>
<td><code>&lt;migrationSource&gt;</code> &lt;brmConfiguration&gt; &lt;brand&gt;BrandOption&lt;/brand&gt; &lt;skipBREMigation&gt;skipOption&lt;/skipBREMigation&gt; &lt;fieldSelection&gt;EventFieldInfo&lt;/fieldSelection&gt; &lt;breConfig&gt; &lt;containerDesc&gt; &lt;param&gt; &lt;paramname&gt;EDRField&lt;/paramname&gt; &lt;paramvalue&gt;EDRFieldValue&lt;/paramvalue&gt; &lt;/param&gt; &lt;/containerDesc&gt; &lt;eventExtension&gt; &lt;param&gt; &lt;paramname&gt;BREEvent&lt;/paramname&gt; &lt;paramvalue&gt;ExtensionBlockName&lt;/paramvalue&gt; &lt;/param&gt; &lt;/eventExtension&gt; &lt;serviceExtension&gt; &lt;param&gt; &lt;paramname&gt;BREService&lt;/paramname&gt; &lt;paramvalue&gt;ExtensionBlockName&lt;/paramvalue&gt; &lt;/param&gt; &lt;/serviceExtension&gt; &lt;defaultValue&gt; &lt;param&gt; &lt;paramname&gt;SERVICE_CLASS&lt;/paramname&gt; &lt;paramvalue&gt;DefaultServiceClass&lt;/paramvalue&gt; &lt;/param&gt; &lt;/defaultValue&gt; &lt;/breConfig&gt; &lt;/brmConfiguration&gt; &lt;/migrationSource&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Contain the details about the source BRM system configuration.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>&lt;brand&gt;</code> element specifies whether the source BRM system supports branding, where BrandOption is:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>enabled</code> to specify that the source BRM system supports branding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>disabled</code> to specify that the source BRM system does not support branding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>&lt;skipBREMigation&gt;</code> element specifies to skip migration of pipeline configuration data, where SkipOption is either true or false.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>&lt;fieldSelection&gt;</code> element contains the information about the target rating engine and the BRM event fields provided as input to the target rating engine for usage rating.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>&lt;breConfig&gt;</code> element contains the batch rating engine configurations.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>&lt;containerDesc&gt;</code> element specifies the EDRC fields that are migrated for events. In the following example, all the EDRC_FIELDS mapped to ALL_RATE are migrated for the events.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;containerDesc&gt; &lt;param&gt; &lt;paramname&gt;EDR_DESC&lt;/paramname&gt; &lt;paramvalue&gt;ALL_RATE&lt;/paramvalue&gt; &lt;/param&gt; &lt;/containerDesc&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>&lt;eventExtension&gt;</code> element specifies to migrate the fields of the specified extension block for the batch rating engine event. Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;eventExtension&gt; &lt;param&gt; &lt;paramname&gt;/event/delayed/session/telco/gsm&lt;/paramname&gt; &lt;paramvalue&gt;ASS_GSM_EXT&lt;/paramvalue&gt; &lt;/param&gt; &lt;/eventExtension&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>&lt;serviceExtension&gt;</code> element specifies to migrate the fields of the specified extension block for the batch rating engine service. Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;serviceExtension&gt; &lt;param&gt; &lt;paramname&gt;/service/telco/gprs&lt;/paramname&gt; &lt;paramvalue&gt;ASS_GPRS_SERV_EXT&lt;/paramvalue&gt; &lt;/param&gt; &lt;/serviceExtension&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>&lt;defaultValue&gt;</code> element specifies the default service class, where DefaultServiceClass is the default service class name.</td>
<td></td>
</tr>
</tbody>
</table>
Table 9–1 (Cont.) Elements in the MigrateConfiguration.xml File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
</table>
| fieldSelection   | <fieldSelection> <targetEngine>TargetRatingEngine</targetEngine> <eventFields> <eventName>EventName</eventName> <fullyQualifiedName>EventFieldName</fullyQualifiedName> </eventFields> </fieldSelection> | Contains the information about the target rating engine and the BRM event fields provided as input to the target rating engine for usage rating.  
  <targetEngine> element specifies the target rating engine used for usage rating, where:  
  TargetRatingEngine is one of the following:  
  - **Convergent Charging**: Specifies that ECE is used for usage rating.  
  - **Realtime Charging**: Specifies that the real-time rating engine is used for usage rating.  
  - **Batch Charging**: Specifies that the batch rating engine is used for usage rating.  
  <eventFields> element specifies the fields that are provided as input to TargetRatingEngine, where:  
  - **eventName** specifies the class name of the BRM event; for example, /event.  
  - **fullyQualifiedName** specifies the fully qualified name of the BRM event field that is provided as input to TargetRatingEngine; for example, PIN_FLD_NAME.  
  Note: Each <fieldSelection> element can have multiple <eventFields> elements but only one <targetEngine> element. To support multiple target rating engines, add the <fieldSelection> element for each target rating engine.  
  Similarly, each <eventFields> element can have multiple <fullyQualifiedName> elements but only one <eventName> element. To rate multiple events, add the <eventFields> element for each event. |
| migrationTarget  | <migrationTarget> <deploymentType>MigrationTargetType</deploymentType> <pricingServer>TargetPDCPricingServer</pricingServer> <brmConfiguration>TargetBRMConfiguration</brmConfiguration> </migrationTarget>   | Contains the details about the migration target system.  
  - **deploymentType** element specifies the migration target, where MigrationTargetType is:  
    PDC_WITH_ONLINE_BATCH_USAGE to specify that the target is a BRM system that uses real-time and batch rating engines and PDC  
    PDC_WITH_CONVERGENT_USAGE to specify that the target is a BRM system that uses ECE and PDC  
  - **pricingServer** element contains the target PDC server information  
  - **brmConfiguration** element contains the target BRM system configuration details. |
**Table 9–1 (Cont.) Elements in the MigrateConfiguration.xml File**

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pricingServer</td>
<td><code>&lt;pricingServer&gt;</code></td>
<td>Contains the target PDC server information, where:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;connectionInfo&gt;</code></td>
<td>■  <code>TargetPricingServerHostName</code> specifies the IP address or the host name of the machine on which the target PDC server is deployed</td>
</tr>
<tr>
<td></td>
<td><code>&lt;hostName&gt;</code> <code>TargetPricingServerHostName</code></td>
<td>■  <code>TargetPricingServerPort</code> specifies the port number of the domain on which the target PDC server is deployed</td>
</tr>
<tr>
<td></td>
<td><code>&lt;port&gt;</code> <code>TargetPricingServerPort</code></td>
<td>■  <code>TargetAdminUserName</code> specifies the user name of the target PDC server administrator</td>
</tr>
<tr>
<td></td>
<td><code>&lt;adminUser&gt;</code> <code>TargetAdminUserName</code></td>
<td>■  <code>TargetPDCUser</code> specifies the user name of the target PDC system user</td>
</tr>
<tr>
<td></td>
<td><code>&lt;pdcUser&gt;</code> <code>TargetPDCUser</code></td>
<td>■  <code>TargetSSLOption</code> specifies whether the target PDC server supports SSL. Enabling SSL enables MigrateBRMPricing to use the <code>t3s://Host:Port</code> URL to access PDC.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;pdcSSL&gt;</code> <code>TargetSSLOption</code></td>
<td>■  Disabling SSL disables SSL. If SSL is disabled, MigrateBRMPricing uses the <code>t3://Host:Port</code> URL to access PDC.</td>
</tr>
<tr>
<td>brmConfiguration</td>
<td><code>&lt;brmConfiguration&gt;</code></td>
<td>Contain the details about the target BRM system configuration, where:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;loadConfigDir&gt;</code> <code>LoadConfigPath</code></td>
<td>■  <code>LoadConfigPath</code> specifies the path to the directory from where the <code>load_config</code> utility is run.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;loadPriceListDir&gt;</code> <code>LoadPriceListPath</code></td>
<td>■  <code>LoadPriceListPath</code> specifies the path to the directory from where the <code>load_price_list</code> utility is run.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;loadIfwConfigDir&gt;</code> <code>LoadPipelineConfigPath</code></td>
<td>■  <code>LoadPipelineConfigPath</code> specifies the path to the directory from where the <code>LoadIfwConfig</code> utility is run.</td>
</tr>
<tr>
<td>logFile</td>
<td><code>&lt;logFile&gt;</code> <code>MigrationLogFileLocation</code></td>
<td>Specifies the directory that stores MigrateBRMPricing log files, where <code>MigrationLogFileLocation</code> is the complete path and the name of the log file.</td>
</tr>
<tr>
<td>reportFile</td>
<td><code>&lt;reportFile&gt;</code> <code>ReportFileLocation</code></td>
<td>Specifies the directory where MigrateBRMPricing stores the reports generated during the migration process, where <code>ReportFileLocation</code> is the complete path to the directory.</td>
</tr>
<tr>
<td>brmExtractedXML</td>
<td><code>&lt;brmExtractedXML&gt;</code> <code>BRMExtractedXML</code></td>
<td>Specifies the directory where MigrateBRMPricing stores the XML files containing the pricing data extracted from the BRM database, where <code>BRMExtractedXML</code> is the complete path to the directory.</td>
</tr>
<tr>
<td>pdcExtractedXML</td>
<td><code>&lt;pdcExtractedXML&gt;</code> <code>PDCExtractedXML</code></td>
<td>Specifies the directory where MigrateBRMPricing stores the XML files containing the pricing data extracted from the PDC database, where <code>PDCExtractedXML</code> is the complete path to the directory.</td>
</tr>
<tr>
<td>brmXML</td>
<td><code>&lt;brmXML&gt;</code> <code>BRMDataFile</code></td>
<td>Specifies the directory where MigrateBRMPricing creates the XML files containing the BRM data that needs to be updated or created in the BRM database, where <code>BRMDataFile</code> is the complete path to the directory.</td>
</tr>
<tr>
<td>pdcXML</td>
<td><code>&lt;pdcXML&gt;</code> <code>PDCDataFile</code></td>
<td>Specifies the directory where MigrateBRMPricing creates the XML files containing the extracted BRM data in PDC format, where <code>PDCDataFile</code> is the complete path to the directory.</td>
</tr>
</tbody>
</table>
Performing Migration of BRM Pricing Data to PDC With ECE

4. Save and close the file.

Migrating the BRM Pricing Data to PDC With ECE

When you migrate pricing data from the BRM database to the PDC database, you must migrate the configuration objects and then migrate the pricing objects.

To migrate the pricing data from the BRM database to the PDC database:

1. Migrate the configuration objects. See "Migrating BRM Configuration Objects".
2. Migrate the pricing objects. See "Migrating BRM Pricing Objects".

Migrating BRM Configuration Objects

To migrate the configuration objects:

1. Ensure that the BRM and PDC databases are running.
2. Go to the BRM_Integration_Pack_Home/apps/migration directory.
3. Run the following command, which starts the data analysis phase:
   
   \texttt{MigrateBRMPricing -config -analyze}

4. Enter the BRM Integration Pack wallet password when prompted.
5. Review the configuration analysis report and fix any data configuration errors that are reported (see "Fixing Errors Found in the Analysis Phase").
6. In step 5, if you made changes to the data in the BRM database, run the following command, which restarts the data analysis phase:
   
   \texttt{MigrateBRMPricing -config -analyze -restart}

7. Restart the ECE server. See the discussion about starting and stopping ECE in BRM Elastic Charging Engine System Administrator’s Guide for more information.
8. Repeat steps 5 to 7 until there are no errors in the configuration analysis report.
9. Run the following command, which starts the data migration phase:
   
   \texttt{MigrateBRMPricing -resume}
10. Review the configuration compatibility report and fix any data configuration errors that are reported (see "Fixing Errors Found in the Migration Phase").

11. In step 10, if you made changes to the data in the BRM database, run the following command, which restarts the data analysis phase:

   `MigrateBRMPricing -restart`

12. Restart the ECE server. See the discussion about starting and stopping ECE in *BRM Elastic Charging Engine System Administrator’s Guide* for more information.

13. Repeat steps 10 to 12 until there are no errors in the configuration compatibility report.

14. Review the migration report and verify that all the configuration objects were migrated to the PDC database successfully.

**Migrating BRM Pricing Objects**

To migrate BRM pricing objects, do one of the following:

- (Recommended) Migrate all pricing objects in one migration process. See "Migrating All BRM Pricing Objects".
- Migrate pricing objects by category. See "Migrating BRM Pricing Objects by Category".
- Migrate pricing objects selectively. See "Migrate Pricing Data Selectively".

**Migrating All BRM Pricing Objects**

To migrate all pricing objects:

1. In the PDC application, ensure that the configuration objects are available.
2. Ensure that the BRM and PDC databases are running.
3. Go to the `BRM_Integration_Pack_Home/apps/migration` directory.
4. Run the following command, which starts the data analysis phase for all pricing objects:

   `MigrateBRMPricing -pricing -analyze`

5. Enter the BRM Integration Pack wallet password when prompted.

6. Review the pricing analysis reports (product, discount, sponsorship and subscription analysis reports) and fix any data configuration errors that are reported (see "Fixing Errors Found in the Analysis Phase").

7. Do one of the following:

   - If you fixed any configuration objects errors reported in the pricing analysis reports, restart the complete migration process. See "Migrating BRM Configuration Objects".
   - If you fixed only pricing objects errors reported in the pricing analysis report, restart the pricing data analysis phase by running the following command:

---

**Note:** `MigrateBRMPricing` does not proceed with the transformation and migration of the configuration objects to the PDC database if there are any critical or user input errors in the configuration analysis report.
Performing Migration of BRM Pricing Data to PDC With ECE

8. Restart the ECE server. See the discussion about starting and stopping ECE in *BRM Elastic Charging Engine System Administrator’s Guide* for more information.

9. Repeat steps 6 to 8 until there are no errors in the pricing analysis reports.

10. Run the following command, which starts the data migration phase:

    `MigrateBRMPricing -resume`

---

**Note:** *MigrateBRMPricing* does not proceed with the transformation and migration of the pricing objects to the PDC database if there are any critical or user input errors in the pricing analysis reports.

11. Review the pricing compatibility reports (product, discount, sponsorship and subscription compatibility reports) and fix any ECE compatibility errors that are reported (see "Fixing Errors Found in the Migration Phase").

12. Do one of the following:

   - If you fixed any configuration object errors reported in the pricing compatibility reports, restart the complete migration process. See "Migrating BRM Configuration Objects".
   - If you fixed only pricing objects errors reported in the pricing compatibility report, restart the pricing data analysis phase by running the following command:

     `MigrateBRMPricing -restart -pricing`

13. Restart the ECE server. See the discussion about starting and stopping ECE in *BRM Elastic Charging Engine System Administrator’s Guide* for more information.

14. Repeat steps 11 to 13 until there are no errors in the pricing compatibility reports.

15. Review the migration report and verify that all pricing objects were migrated to PDC successfully.

16. Make corrections to PDC data, if needed. See "Changes Required After Migration".

**Migrating BRM Pricing Objects by Category**

When you migrate pricing objects by category, the migration of all the objects in each category must be completed before starting the next.

To migrate pricing objects by category:

1. In the PDC application, ensure that the configuration objects are available.

2. Ensure that the BRM and PDC databases are running.

3. Go to the *BRM_Integration_Pack_Home/apps/migration* directory.

4. Run the following command, which starts the data analysis phase for products and all objects referenced by a product:

   `MigrateBRMPricing -product -analyze`

5. Enter the BRM Integration Pack Wallet password when prompted.
6. Review the product analysis report and fix any data configuration errors that are reported (see "Fixing Errors Found in the Analysis Phase").

7. Do one of the following:
   - If you fixed any configuration object errors reported in the product analysis report, restart the complete migration process. See "Migrating BRM Configuration Objects".
   - If you fixed only pricing objects errors reported in the product analysis report, restart the data analysis phase for products by running the following command:
     ```
     MigrateBRMPricing -product -analyze -restart
     ```

8. Restart the ECE server. See the discussion about starting and stopping ECE in *BRM Elastic Charging Engine System Administrator’s Guide* for more information.

9. Repeat steps 6 to 8 until there are no errors in the product analysis report.

10. Run the following command, which starts the data migration phase for products:
    ```
    MigrateBRMPricing -resume
    ```

    **Note:** *MigrateBRMPricing* does not proceed with the transformation and migration of the pricing objects to the PDC database if there are any critical or user input errors in the analysis reports.

11. Review the product compatibility report and fix any data configuration errors that are reported (see "Fixing Errors Found in the Migration Phase").

12. Do one of the following:
    - If you fixed any configuration object errors reported in the product compatibility report, restart the complete migration process. See "Migrating BRM Configuration Objects".
    - If you fixed only pricing objects errors reported in the product compatibility report, restart the data analysis phase for products by running the following command:
      ```
      MigrateBRMPricing -restart -product
      ```

13. Restart the ECE server. See the discussion about starting and stopping ECE in *BRM Elastic Charging Engine System Administrator’s Guide* for more information.

14. Repeat steps 11 to 13 until there are no errors in the product compatibility report.

15. Review the migration report and verify that the products and objects referenced by a product were migrated to PDC successfully.

16. Make corrections to PDC data, if needed. See "Changes Required After Migration".

17. Run the following command, which starts the data analysis phase for discounts and all objects referenced by a discount:
    ```
    MigrateBRMPricing -discount -analyze
    ```

18. Enter the BRM Integration Pack wallet password when prompted.
19. Review the discount analysis report and fix any data configuration errors that are reported.

20. Do one of the following:

- If you fixed any configuration object errors reported in the discount analysis report, restart the complete migration process. See "Migrating BRM Configuration Objects".
- If you fixed only pricing objects errors reported in the discount analysis report, restart the data analysis phase for discounts by running the following command:

  `MigrateBRMPricing -discount -analyze -restart`

21. Restart the ECE server. See the discussion about starting and stopping ECE in 

22. Repeat steps 19 to 21 until there are no errors in the discount analysis report.

23. Run the following command, which starts the data migration phase for discounts:

  `MigrateBRMPricing -resume`

24. Review the discount compatibility report and fix any data configuration errors that are reported.

25. Do one of the following:

- If you fixed any configuration object errors reported in the discount compatibility report, restart the complete migration process. See "Migrating BRM Configuration Objects".
- If you fixed only pricing objects errors reported in the discount compatibility report, restart the data analysis phase for discounts by running the following command:

  `MigrateBRMPricing -restart -discount`

26. Restart the ECE server. See the discussion about starting and stopping ECE in 

27. Repeat steps 24 to 26 until there are no errors in the discount compatibility report.

28. Review the migration report and verify that the discounts and all objects referenced by a discount were migrated to PDC successfully.

29. Make corrections to PDC data, if needed. See "Changes Required After Migration".

30. Run the following command, which starts the data analysis phase for chargeshares and all objects referenced by a chargeshare:

  `MigrateBRMPricing -sponsorship -analyze`

31. Enter the BRM Integration Pack wallet password when prompted.

32. Review the sponsorship analysis report and fix any data configuration errors that are reported.

33. Do one of the following:

- If you fixed any configuration object errors reported in the sponsorship analysis report, restart the complete migration process. See "Migrating BRM Configuration Objects".
If you fixed only pricing objects errors reported in the sponsorship analysis report, restart the data analysis phase for chargeshares by running the following command:

```
MigrateBRMPricing -sponsorship -analyze -restart
```

34. Restart the ECE server. See the discussion about starting and stopping ECE in BRM Elastic Charging Engine System Administrator’s Guide for more information.

35. Repeat steps 32 to 34 until there are no errors in the sponsorship analysis report.

36. Run the following command, which starts the data migration phase for chargeshares:

```
MigrateBRMPricing -resume
```

37. Review the sponsorship compatibility report and fix any data configuration errors that are reported.

38. Do one of the following:
   - If you fixed any configuration object errors reported in the sponsorship compatibility report, restart the complete migration process. See "Migrating BRM Configuration Objects".
   - If you fixed only pricing objects errors reported in the sponsorship compatibility report, restart the data analysis phase for chargeshares by running the following command:

```
MigrateBRMPricing -restart -sponsorship
```

39. Restart the ECE server. See the discussion about starting and stopping ECE in BRM Elastic Charging Engine System Administrator’s Guide for more information.

40. Repeat steps 37 to 39 until there are no errors in the sponsorship analysis report.

41. Review the migration report and verify that the chargeshares and all objects referenced by a chargeshare were migrated to PDC successfully.

42. Make corrections to PDC data, if needed. See "Changes Required After Migration".

43. Run the following command, which starts the data analysis phase for subscription objects:

```
MigrateBRMPricing -subscription -analyze
```

44. Enter the BRM Integration Pack wallet password when prompted.

45. Review the subscription analysis report and fix any data configuration errors that are reported.

46. Do one of the following:
   - If you fixed any configuration object errors reported in the subscription analysis report, restart the complete migration process. See "Migrating BRM Configuration Objects".
   - If you fixed only pricing objects errors reported in the subscription analysis report, restart the data analysis phase for subscription objects by running the following command:

```
MigrateBRMPricing -subscription -analyze -restart
```

47. Restart the ECE server. See the discussion about starting and stopping ECE in BRM Elastic Charging Engine System Administrator’s Guide for more information.
48. Repeat steps 45 to 47 until there are no errors in the subscription analysis report.

49. Run the following command, which starts the data migration phase for subscription objects:

   `MigrateBRMPricing -resume`

50. Review the sponsorship compatibility report and fix any data configuration errors that are reported.

51. Do one of the following:

   - If you fixed any configuration object errors reported in the subscription compatibility report, restart the complete migration process. See "Migrating BRM Configuration Objects".
   - If you fixed only pricing objects errors reported in the subscription compatibility report, restart the data analysis phase for subscription by running the following command:

     `MigrateBRMPricing -restart -subscription`

52. Restart the ECE server. See the discussion about starting and stopping ECE in BRM Elastic Charging Engine System Administrator’s Guide for more information.

53. Repeat steps 50 to 52 until there are no errors in the sponsorship compatibility report.

54. Review the migration report and verify that all subscription objects were migrated to PDC successfully.

55. Make corrections to PDC data, if needed. See "Changes Required After Migration".

56. Using the target PDC application, verify that you are able to create and modify subscription objects without any errors.

---

**Migrate Pricing Data Selectively**

You can migrate the following pricing objects selectively to PDC: products, discounts, and sponsorships.

---

**Note:** You cannot migrate subscriptions (bundles or packages) selectively.

You cannot migrate data incrementally. After migrating a set of pricing objects, restart the migration process with new set of pricing objects.

---

To migrate the pricing objects selectively:

1. In the PDC application, ensure that the configuration objects are available.

2. Ensure that the BRM and PDC databases are running.

3. Go to the `BRM_Integration_Pack_Home/apps/migration` directory.

4. Create an XML file with the required pricing objects to be migrated; for example, `pricingbatch.xml`.

   This XML file can include either the services for which the pricing objects have to be migrated (for example, `/service/telco/gsm/gprs`) or the categories of pricing objects (for example, product) that need to be migrated. Multiple pricing objects can be separated by a comma.
Pricing objects for both the specific service and the specific category cannot be migrated together. You can migrate pricing objects either by services associated with these objects or by the category of pricing objects, selectively, at a time. If both service types and offering codes are specified in the same XML file, offering codes takes precedence.

The following example shows the sample XML file created for migrating pricing objects by services:

```xml
<!--Example for selectively migrating services. Note that services are listed under the name tag. -->
<filterBy>
  <service>
    <name>/service/telco/gsm/roaming</name>
    <name>/service/telco/gsm/gprs</name>
    <name>/service/subscription</name>
  </service>
</filterBy>
```

The following example shows the sample XML file created for migrating pricing objects by category:

```xml
<!--Example for selectively migrating Pricing objects. Note that pricing Objects are comma separated. -->
<filterBy>
  <product>
    <codes>Product1,Product2</codes>
  </product>
  <discount>
    <codes>Discount1,Discount2</codes>
  </discount>
  <sponsorship>
    <codes>Sponsorship1,Sponsorship2</codes>
  </sponsorship>
</filterBy>
```

5. Continue the pricing objects migration process. See "Migrating BRM Pricing Objects by Category".

About Migrating PDC Pricing Data to PDC With ECE

If you are using PDC to create and manage your price lists and using the real-time and batch rating engines for usage rating, and you want to use ECE for usage rating, you must first install PDC with ECE and then migrate the pricing data from the source PDC database into the target PDC database.

You use the command-line utility, `MigrateBRMPricing`, to migrate the PDC pricing data into the target PDC database.

Migration of the PDC pricing data is performed in a single phase, the data migration phase. In the data migration phase, the migration utility extracts the existing PDC data, checks for ECE compatibility, adjusts the pricing data, and migrates the adjusted data to the target PDC database.

You must perform the migration process twice. First you migrate configuration objects. Then, you migrate pricing objects.

The migration utility can adjust most pricing configurations. Adjustments to the PDC pricing data configurations may result in existing objects being updated or new objects
being created. In some cases, the migration utility may require additional information from you before it can adjust the data. But, there are some configurations that it cannot adjust and for those cases, you may need to change the configurations manually before continuing with the migration.

You run the `MigrateBRMPricing` or `MigrateBRMPricing -resume` command to start the data migration phase.

**About the Data Migration Phase for Migrating PDC Pricing Data to PDC With ECE**

The following procedure describes how the PDC pricing data is processed in the migration phase:

1. The migration utility extracts the PDC pricing data and saves the data in XML files.

2. The migration utility checks the data to determine if the configurations are supported in ECE and whether any configurations need to be adjusted prior to the migration.

3. The migration utility generates compatibility reports that provide the list of the data configurations that need to be adjusted by the utility during the migration process and any configurations that you would need to adjust manually.

   You need to review the compatibility reports, resolve any pricing data compatibility errors in the PDC database, and restart the migration process.

   When you restart the migration process, the compatibility reports are also re-generated for the latest data.

4. The migration utility loads the PDC components into the target PDC database.

5. The migration utility generates the migration report with information about the PDC objects that were successfully migrated to the PDC database.

For more information about the migration utility reports, see "About Migration Reports".

**Figure 9–3** shows the data migration phase for migrating PDC data to PDC with ECE:
Performing Migration of PDC Pricing Data to PDC With ECE

To set up and migrate the PDC pricing data to PDC with ECE:

1. Create a backup of the PDC production database.
   See "Creating a BackUp of Your BRM Database".

2. Setup the PDC development system to run the initial migration of the pricing data.
   See "Setting Up the BRM Development System" for more information.

3. Migrate the PDC configuration and pricing objects.
   See "Migrating the PDC Pricing Data to PDC With ECE" for more information.

4. Restart the PDC and ECE systems.
   See "Restarting the Systems".

5. Test the migrated PDC data.
   See "Testing the Migrated Data".

6. Import the migrated data to the PDC production systems.
   See "Importing the Migrated Data to the Production Systems".

Creating a BackUp of Your PDC Database

Oracle recommends that you perform a full backup of your PDC production database prior to running the migration.
Performing a full backup allows you to restore your production database to its original state in the event of any data corruption or data loss. Back up both the database definition and all the database contents.

Use the backup to restore the database on a test system to ensure the backup is valid. See your database software documentation for more information on performing full database backups.

**Setting Up the PDC Development System**

Setting up the PDC development system includes installing a PDC system. Perform the following tasks to setup the PDC development system:

1. Review the system requirements for installing the PDC system.
   
   See the discussion about Pricing Design Center system requirements in *PDC Installation and System Administration Guide*.

2. Install the complete PDC software.

   **Important:** Before you install PDC, ensure that the PDC database does not have any existing pricing data.

   When you install BRM Integration Pack, ensure that you select the **Migrate PDC Pricing Data to PDC with ECE** option.

3. Copy the configuration objects from the BRM database to the development system by using BRM load utilities. See the discussion about the type of configuration objects (setup components in PDC) that are mastered in BRM and their corresponding load utilities in *PDC Creating Product Offerings*.

4. Copy the PDC pricing data from the production database to the development system.

   You can copy the PDC pricing data in one of two ways:

   - Use the **ImportExportPricing** utility to export the pricing data from your PDC production database and load it into the PDC development database.
     
     See the discussion about the **ImportExportPricing** utility in *PDC Creating Product Offerings* for more information.
   
   - Restore the database from the BRM database backup.
     
     See your database software documentation for more information about database restore.

5. Ensure **BRM_Integration_Pack_Home/apps/bin** is in your PATH environment variable, where **BRM_Integration_Pack_Home** is the directory in which you installed BRM Integration Pack.

6. (Optional) Configure the **MigrateBRMPricing** utility. See "Configuring the Migration Utility" for more information.

7. Ensure all users of the migration utility are added to the **Migration Admin** group.
The users of the migration utility must belong to the **Migration Admin** group, which is created during the PDC installation. You can add additional users to this group by using the WebLogic Administration Console.

8. Obtain the Java Keystore password.

**Migrating the PDC Pricing Data to PDC With ECE**

When you migrate pricing data from the source PDC database to the target PDC database, you must migrate the configuration objects and then migrate the pricing objects.

To migrate the pricing data from the source PDC database to the target PDC database:

1. Migrate the configuration objects. See "Migrating BRM Configuration Objects".
2. Migrate the pricing objects. See "Migrating BRM Pricing Objects".

**Migrating PDC Configuration Objects**

To migrate the configuration objects:

1. Ensure that the BRM and PDC databases are running.
2. Go to the `BRM_Integration_Pack_Home/apps/migration` directory.
3. Run the following command, which starts the data migration phase:
   ```
   MigrateBRMPricing -config
   ```
4. Enter the BRM Integration Pack wallet password when prompted.
5. Review the configuration compatibility report and fix any data configuration errors that are reported (see "Fixing Errors Found in the Migration Phase").
6. In step 5, if you made changes to the data in the BRM database, run the following command, which restarts the data migration phase:
   ```
   MigrateBRMPricing -resume
   ```
7. Repeat steps 5 and 6 until there are no errors in the configuration compatibility report.
8. Review the migration report and verify that all the configuration objects were migrated to the target PDC system successfully.
9. Using the target PDC application, verify that you are able to view, create, and modify configuration objects without any errors.

**Migrating PDC Pricing Objects**

To migrate PDC pricing objects, do one of the following:

- (Recommended) Migrate all pricing objects in one migration process. See "Migrating All PDC Pricing Objects".
- Migrate pricing objects by category. See "Migrating PDC Pricing Objects by Category".

**Migrating All PDC Pricing Objects**

To migrate all pricing objects:

1. In the target PDC application, ensure that the configuration objects are available.
2. Ensure that the BRM and PDC databases are running.
3. Go to the `BRM_Integration_Pack_Home/apps/migration` directory.

4. Run the following command, which starts the data migration phase for all pricing objects:

   ```
   MigrateBRMPricing -pricing
   ```

5. Enter the BRM Integration Pack wallet password when prompted.

6. Review the pricing compatibility reports (product, discount, sponsorship and subscription analysis reports) and fix any data configuration errors that are reported. If you encounter any critical errors while migrating pricing objects from BRM to PDC, run the `MigrateBRMPricing` utility with `-skip` parameter to skip critical errors and continue the migration process (see "Fixing Errors Found in the Migration Phase").

   **Important:** You can use the `-skip` parameter only if you are migrating BRM pricing objects to PDC with ECE.

   Critical errors encountered due to configuration objects cannot be skipped by using the `-skip` parameter. You need to manually fix the errors and then restart the migration process.

7. Do one of the following:
   - If you fixed any configuration object errors reported in the pricing compatibility reports, restart the complete migration process. See "Migrating Pricing Data to PDC With ECE".
   - If you fixed only pricing objects errors reported in the pricing compatibility reports, restart the pricing data migration phase by running the following command:
     ```
     MigrateBRMPricing -restart -pricing
     ```

8. Restart the ECE server. See the discussion about starting and stopping ECE in `BRM Elastic Charging Engine System Administrator’s Guide` for more information.

9. Repeat steps 6 to 8 until there are no errors in the pricing compatibility reports.

   **Note:** `MigrateBRMPricing` does not proceed with the migration of the pricing objects to the target PDC database if there are any critical or user input errors in the pricing compatibility reports.

10. Review the migration report and verify that all pricing objects were migrated to the target PDC system successfully.

11. Make corrections to data in the target PDC system, if needed. See "Changes Required After Migration".

12. Using the target PDC application, verify that you are able to create and modify pricing objects without any errors.

**Migrating PDC Pricing Objects by Category**

When you migrate pricing objects by category, the migration of all the objects in each category must be completed before starting the next.

To migrate pricing objects by category:
Performing Migration of PDC Pricing Data to PDC With ECE

1. In the target PDC application, ensure that the configuration objects are available.
2. Ensure that the BRM and PDC databases are running.
3. Go to the BRM_Integration_Pack_Homeapps/migration directory.
4. Run the following command, which starts the data analysis phase for products and all objects referenced by a product:
   
   MigrateBRMPricing -product

5. Enter the BRM Integration Pack wallet password when prompted.
6. Review the product compatibility report and fix any data configuration errors that are reported (see "Fixing Errors Found in the Migration Phase").
7. Do one of the following:
   - If you fixed any configuration object errors reported in the product compatibility report, restart the complete migration process. See "Migrating Pricing Data to PDC With ECE".
   - If you fixed only pricing objects errors reported in the product compatibility report, restart the data migration phase for products by running the following command:
     
     MigrateBRMPricing -restart -product

   - If you encounter any critical errors while migrating pricing objects from BRM to PDC, run the MigrateBRMPricing utility with -skip parameter to skip critical errors and continue the migration process. See "Skipping Errors and Continue Pricing Objects Migration".
8. Restart the ECE server. See the discussion about starting and stopping ECE in BRM Elastic Charging Engine System Administrator's Guide for more information.
9. Repeat steps 6 to 8 until there are no errors in the product compatibility report.

---

**Note:** MigrateBRMPricing does not proceed with the migration of the pricing objects to the target PDC database if there are any critical or user input errors in the compatibility reports.

10. Review the migration report and verify that the products and objects referenced by a product were migrated to the target PDC system successfully.
11. Make corrections to the target PDC data, if needed. See "Changes Required After Migration".
12. Using the target PDC application, verify that you are able to create and modify products without any errors.
13. Run the following command, which starts the data migration phase for discounts and all objects referenced by a discount:
   
   MigrateBRMPricing -discount

14. Enter the BRM Integration Pack wallet password when prompted.
15. Review the discount compatibility report and fix any data configuration errors that are reported.
16. Do one of the following:
If you fixed any configuration object errors reported in the discount compatibility report, restart the complete migration process. See "Migrating Pricing Data to PDC With ECE".

If you fixed only pricing objects errors reported in the discount compatibility report, restart the data migration phase for discounts by running the following command:

```
MigrateBRMPricing -restart -discount
```

17. Restart the ECE server. See the discussion about starting and stopping ECE in BRM Elastic Charging Engine System Administrator's Guide for more information.

18. Repeat steps 15 to 17 until there are no errors in the discount compatibility report.

19. Review the migration report and verify that the discounts and all objects referenced by a discount were migrated to the target PDC system successfully.

20. Make corrections to data in the target PDC system, if needed. See "Changes Required After Migration".

21. Using the target PDC application, verify that you are able to create and modify discounts without any errors.

22. Run the following command, which starts the data migration phase for chargeshares and all objects referenced by a chargeshare:

```
MigrateBRMPricing -sponsorship
```

23. Enter the BRM Integration Pack wallet password when prompted.

24. Review the sponsorship analysis report and fix any data configuration errors that are reported.

25. Do one of the following:

   - If you fixed any configuration object errors reported in the sponsorship compatibility report, restart the complete migration process. See "Migrating Pricing Data to PDC With ECE".

   - If you fixed only pricing objects errors reported in the sponsorship compatibility report, restart the data migration phase for chargeshares by running the following command:

     ``
     MigrateBRMPricing -restart -sponsorship
     ``

26. Restart the ECE server. See the discussion about starting and stopping ECE in BRM Elastic Charging Engine System Administrator's Guide for more information.

27. Repeat steps 24 to 26 until there are no errors in the sponsorship analysis report.

28. Review the migration report and verify that the chargeshares and all objects referenced by a chargeshare were migrated to the target PDC system successfully.

29. Make corrections to data in the target PDC system, if needed. See "Changes Required After Migration".

30. Run the following command, which starts the data migration phase for subscription objects:

```
MigrateBRMPricing -subscription
```

31. Enter the BRM Integration Pack wallet password when prompted.
32. Review the subscription compatibility report and fix any data configuration errors that are reported.

33. Do one of the following:
   - If you fixed any configuration object errors reported in the subscription compatibility report, restart the complete migration process. See "Migrating Pricing Data to PDC With ECE".
   - If you fixed only pricing objects errors reported in the subscription compatibility report, restart the data migration phase for subscriptions by running the following command:
     ```
     MigrateBRMPricing --restart -subscription
     ```

34. Restart the ECE server. See the discussion about starting and stopping ECE in BRM Elastic Charging Engine System Administrator’s Guide for more information.

35. Repeat steps 32 to 34 until there are no errors in the subscription analysis report.

36. Review the migration report and verify that all subscription objects were migrated to the target PDC system successfully.

37. Make corrections to data in the target PDC system, if needed. See "Changes Required After Migration".

38. Using the target PDC application, verify that you are able to create and modify subscription objects without any errors.

Fixing Errors Found in the Analysis Phase

The following sections describe how to provide required input data and how to fix some of the common critical errors reported in the migration analysis reports. The migration analysis report lists the data configurations that are not supported in PDC.

Fixing Analysis "User Input Required" Errors

The migration utility may require you to provide additional information prior to migrating resources and discount and chargeshare configurations from the BRM database to the PDC database. The analysis reports list these objects marked as User Input Required with a description of the information that is required. You provide the input in an XML file.

To provide the input:

1. Make a copy of the predefined XML templates in the `BRM_Integration_Pack_Home/apps/xml` directory in the `BRM_Integration_Pack_Home/apps/xml/userinput` directory, where `BRM_Integration_Pack_Home` is the directory in which the PDC software is installed.

2. Update the XML file content with the required information.

3. Ensure that the XML input file conforms to the XML schema definition in the `BRM_Integration_Pack_Home/apps/xsd` directory.

See the following topics for more information:

- Identifying Resource Id for Non-Currency Resources Used in Batch Rating
- Providing Values List for Expressions in Discount and Chargeshare Configurations
Identifying Resource Id for Non-Currency Resources Used in Batch Rating

If you have a non-currency resource that is used in batch rating, you need to map the resource name to the resource Id defined in the BRM server so that the resource can be mapped to the BEID definition.

The following is the XML that you would provide to map the resource LOYALTY to resource Id 1000019.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Resource>
  <BalanceElement type="0">
    <BREResourceName>LOYALTY</BREResourceName>
    <BEIDCode>LOY</BEIDCode>
    <ResId>1000019</ResId>
  </BalanceElement>
</Resource>
```

Providing Values List for Expressions in Discount and Chargeshare Configurations

If you use an expression as a filter criteria in the discount and chargeshare detail, you need to list the values in the expression individually.

**Example 1**

If you use the expression Peak* with Peak1, Peak2, and Peak3 for time model and 21* with 2155557 and 2155558 for resource Id in the discount detail, you need to provide these values individually. Following is an example of the XML file that you would provide:

```xml
<Filters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="oracle\communications\brm\pdc\migration\pricing\impl\bre\userinputxml\BreFilter_UserInput.xsd">
  <Filter master='M1' rank='1'>
    <TimeModel>
      <name>Peak1</name>
      <name>Peak2</name>
      <name>Peak3</name>
    </TimeModel>
    <ResID>
      <id>2155557</id>
      <id>2155558</id>
    </ResID>
  </Filter>
</Filters>
```

Alternatively, you can correct the expression in the discount detail itself. If you change the discount detail configuration, you need to restart the migration utility.

**Example 2**

If you specify an expression in the rate plan field, such as !(Rate1 | Rate2 | Rate3), you need to provide the XML that specifies the rate plans that should apply rather than those that should be excluded. This is because the NOT operator is not supported in PDC.

Analysis Errors Requiring Pricing Data Changes

The following are some common BRM configurations that require manual changes prior to migrating the data to the PDC data model. These configurations are listed in the migration analysis reports and the report also specifies exactly where the issues
reside in your data. They are listed here for your reference and to provide more details on how to resolve them.

The following changes are required prior to migration:

- Multiple Impact Categories in a Rate Plan
- Derived Impact Category Used in Zone Model
- Zone Items Differentiated Only by Service Code
- Extended Service Class Event Mapping
- Service Contains Deal Mapped to Parent Service
- Using Overlapping Date Ranges to Sequentially Impact Multiple Resources
- Quantity Brackets in a Fold
- BRM Object Names Containing Special Characters
- Custom Fields

Multiple Impact Categories in a Rate Plan

In PDC, the balance impact in charge does not contain impact categories. If you migrate a BRM rate plan selector with multiple rules which result in the same rate plan but with different impact categories, the migration utility reports an error. If you migrate a rate plan selector with multiple rules that result in the same rate plan with the same impact category, this rate plan selector is migrated as long as the impact category is specified in the rateplan.

Recommended Action

For the rate plan selector which has multiple rules that result in the same rate plan with different impact categories, create a separate rate plan for each impact category. In each newly created rate plan, inspect the rate objects and remove any balance impacts that specify impact categories not applicable to that rate plan.

Example 1

Table 9–2 shows a rate plan selector with multiple rules that result in the same rate plan (rate plan A) with different impact categories:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Rule Expression</th>
<th>Rate Plan</th>
<th>Impact Category Used in Rate Plan Selector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1</td>
<td>Attribute values</td>
<td>Rate plan A</td>
<td>IC1</td>
</tr>
<tr>
<td>Rule 2</td>
<td>Attribute values</td>
<td>Rate plan A</td>
<td>IC2</td>
</tr>
<tr>
<td>Rule 3</td>
<td>Attribute values</td>
<td>Rate plan A</td>
<td>*</td>
</tr>
<tr>
<td>Rule 4</td>
<td>Attribute values</td>
<td>Rate plan A</td>
<td>default</td>
</tr>
</tbody>
</table>

Table 9–3 shows the rate objects in rate plan A.

<table>
<thead>
<tr>
<th>Rate Object</th>
<th>Impact Category Used in Rate Object</th>
<th>Balance Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Object 1</td>
<td>IC1</td>
<td>$1.00</td>
</tr>
</tbody>
</table>
Fixing Errors Found in the Analysis Phase

Migrating Pricing Data to PDC With ECE

Solution

Make the changes shown in Table 9–4 to the rate objects:

Table 9–4  Rate Plan Selector

<table>
<thead>
<tr>
<th>Rule</th>
<th>Rule Expression</th>
<th>Rate Plan</th>
<th>Impact Category Used in Rate Plan Selector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1</td>
<td>Attribute values</td>
<td>Rateplan A_IC1</td>
<td>IC1</td>
</tr>
<tr>
<td>Rule 2</td>
<td>Attribute values</td>
<td>Rateplan A_IC2</td>
<td>IC2</td>
</tr>
<tr>
<td>Rule 3</td>
<td>Attribute values</td>
<td>Rateplan A_*</td>
<td>*</td>
</tr>
<tr>
<td>Rule 4</td>
<td>Attribute values</td>
<td>Rateplan A_default</td>
<td>default</td>
</tr>
</tbody>
</table>

Table 9–5 shows the rate objects in rate plan A_IC1.

Table 9–5  Rate Plan A_IC1

<table>
<thead>
<tr>
<th>Rate Object</th>
<th>Impact Category Used in Rate Object</th>
<th>Balance Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Object 1</td>
<td>IC1</td>
<td>$1.00</td>
</tr>
<tr>
<td></td>
<td>IC1 (previously *)</td>
<td>$5.00</td>
</tr>
<tr>
<td>Rate Object 2</td>
<td>IC1</td>
<td>$1.50</td>
</tr>
<tr>
<td></td>
<td>IC1 (previously *)</td>
<td>$5.00</td>
</tr>
<tr>
<td>Rate Object 3</td>
<td>IC1</td>
<td>$0.50</td>
</tr>
<tr>
<td>Rate Object 4</td>
<td>IC1 (previously default)</td>
<td>$1.00</td>
</tr>
</tbody>
</table>

Table 9–6 shows the rate objects in rate plan A_IC2.

Table 9–6  Rate Plan A_IC2

<table>
<thead>
<tr>
<th>Rate Object</th>
<th>Impact Category Used in Rate Object</th>
<th>Balance Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Object 1</td>
<td>IC2</td>
<td>$2.00</td>
</tr>
<tr>
<td></td>
<td>IC2 (previously *)</td>
<td>$5.00</td>
</tr>
</tbody>
</table>
Fixing Errors Found in the Analysis Phase

Table 9–7 shows the rate objects in rate plan A_IC3.

<table>
<thead>
<tr>
<th>Rate Object</th>
<th>Impact Category Used in Rate Object</th>
<th>Balance Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Object 2</td>
<td>IC2 (previously *)</td>
<td>$6.00</td>
</tr>
<tr>
<td>Rate Object 3</td>
<td>IC2 (previously default)</td>
<td>$0.50</td>
</tr>
<tr>
<td>Rate Object 4</td>
<td>IC2 (previously default)</td>
<td>$1.00</td>
</tr>
</tbody>
</table>

Table 9–8 shows the rate objects in rate plan A_default.

<table>
<thead>
<tr>
<th>Rate Object</th>
<th>Impact Category Used in Rate Object</th>
<th>Balance Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Object 1</td>
<td>default</td>
<td>$0.50</td>
</tr>
<tr>
<td></td>
<td>default (previously *)</td>
<td>$5.00</td>
</tr>
<tr>
<td>Rate Object 2</td>
<td>default (previously *)</td>
<td>$6.00</td>
</tr>
<tr>
<td>Rate Object 3</td>
<td>default</td>
<td>$0.50</td>
</tr>
<tr>
<td>Rate Object 4</td>
<td>default</td>
<td>$0.50</td>
</tr>
</tbody>
</table>

Derived Impact Category Used in Zone Model

Impact categories defined for zoning can be flagged as derived, which means that they can only be used in USC and APN selectors. BRM allows derived impact categories to be used in a zone model, but PDC does not. If you migrate a BRM zone model with an impact category of type derived, the migration utility reports an error.

**Recommended Action**

If a derived impact category is used in a zone model, change the type from 1 (derived) to 0.

**Example**

The impact category, Brazil, which is associated with the zone item, San Jose to Brazil, is defined as derived and is used in a zone model.

**Solution**

Update the IFW_IMPACT_CAT table using the following SQL statement:

```
UPDATE IFW_IMPACT_CAT SET TYPE=0 WHERE Impact_Category LIKE 'Brazil';
```
Zone Items Differentiated Only by Service Code
In PDC, a zone item in a zone model does not specify a service code. In BRM, if two zone items in a zone model differ by only the service code, then the migrated zone items gives a unique constraint error in PDC because they will be identical.

Recommended Action
Modify the zone items that differ only by service code to make them unique or remove the duplicate item.

Example

<table>
<thead>
<tr>
<th>Origin</th>
<th>Destination</th>
<th>Valid From</th>
<th>Service</th>
<th>Zone Impact Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>456</td>
<td>1999-01-01</td>
<td>SMS1</td>
<td>IC1</td>
</tr>
<tr>
<td>123</td>
<td>456</td>
<td>1999-01-01</td>
<td>SMS2</td>
<td>IC2</td>
</tr>
</tbody>
</table>

Solution
Modify the ValidFrom field in the IFW_STANDARD_ZONE table to a unique value (for example, 02-JAN-99 in the above example), to make the combination unique without using the service code.

Extended Service Class Event Mapping
In PDC, an extended service class does not inherit the event mapping of the parent class, and all events to be used with the extended service class must be explicitly mapped. In BRM, an extended service class inherits the event mapping of the parent class. If you migrate a BRM pricing object with a service-event combination that is not available in the service-event mapping, the migration utility reports an error.

Recommended Action
Ensure that all extended service classes have mappings for all the events used in pricing objects that reference them.

Example
The service-event map (pin_event_map file) contains these mappings:

```
/service/ip   :/event/session   ...
  :/event/session/dialup ...
/service/ip/gprs  :/event/session/gprs/master
  :/event/session/gprs/subsession ...
```

A product exists that applies to /service/ip/gprs and /event/session. The migration utility reports an error because /event/session is not mapped to /service/ip/gprs.

Solution
Map /service/ip/gprs to /event/session.

```
/service/ip   :/event/session   ...
  :/event/session/dialup ...
/service/ip/gprs  :/event/session/gprs/master
  :/event/session/gprs/subsession ...
  :/event/session
```
Service Contains Deal Mapped to Parent Service
In BRM, Pricing Center allows to add a service to a plan and to add a deal to that service which applies to a parent class of the service. PDC does not allow this configuration, and the migration utility reports an error if such a plan is migrated.

Recommended Action
Change the service in the plan to match the service referenced in the deal.

Example
Plan A applies to /service/telco/gsm/voice and contains:

- Deal 1, which applies to /service/telco.
- Deal 2, which applies to /service/telco/gsm.

Solution
The plan should be changed to have the following service/deal combination:

- /service/telco
  - Deal 1
- /service/telco/gsm
  - Deal 2

Using Overlapping Date Ranges to Sequentially Impact Multiple Resources
PDC does not support BRM rate plans with more than one date tier with overlapping date ranges. This type of configuration is typically used to first consume an included balance, such as Anytime Minutes and after the balance is exhausted, to charge for the remaining usage.

Recommended Action
Re-configure the product to contain only a usage charge and configure a discount to credit the charge and debit the available balance.

Example
A rate plan for voice calls has two date tiers:

- Consume Minutes, with a validity period that starts immediately and never ends.
- Charges for call, with a validity period that starts immediately and never ends.

Solution
Re-configure the product by doing the following:
1. Create a discount based on the user scenario for consumption.
2. Remove the consumption tier from the product.
3. Do one of the following:
   - Create a new deal and add the discount created in step 1.
   - Add the discount created in step 1 to the existing deal and cancel the existing deal for the subscriber.
4. Purchase the new or modified deal in BRM.
Quantity Brackets in a Fold

PDC requires the basis for quantity brackets in a fold to be specified as Resource Balance and the resource to be a non-currency resource. If you migrate a BRM fold which has basis for quantity bracket configured with anything other than Resource Balance or the resource is a currency resource, the migration utility reports an error.

Recommended Action

Correct the fold configuration to use Resource Balance as the basis for quantity brackets and specify a non-currency resource.

Examples

- The basis for quantity discount bracket is Continuous or Rate Dependent.
- The basis is Resource Balance but the resource is a currency resource.

Solution

Review the business scenario and ensure that your fold configuration is implemented correctly. You can only fold a non-currency resource and the basis for quantity brackets in a fold must be Resource Balance.

BRM Object Names Containing Special Characters

PDC does not support the character "=" in a pricing object name. The migration utility reports an error if such objects are migrated.

Recommended Action

Change the pricing object name to not contain the "=" character.

Custom Fields

The migration utility is not aware of any custom fields that you may have created in the BRM database.

For example, the migration utility reports the following error while converting the event storable class to XML if the event storable class has custom fields:

SEVERE: Error while run migration!
oracle.communications.brm.pdc.migration.MigrationException: Got a non zero exit status from the command:
/home/pin/7.5/bin/storableclasstoxml -r
/home/pin/opt/oracle/11.2.0.3/apps/migration/xml/extract/Event.xml -o /event/*.
Check if the BRM is up and running.

Recommended Action

Make custom fields available to applications. Refer to the discussion about making custom fields available to your applications for Java application in BRM Developer’s Guide.

Fixed Errors Found in the Migration Phase

During the data migration phase, the PDC data is validated against pricing features supported in ECE. The compatibility reports contain lists of errors that need to be fixed.
Fixing Errors Found in the Migration Phase

prior to completing the migration. The following sections describe how to provide user input and how to fix the most common critical errors in the compatibility reports.

**Skipping Errors and Continue Pricing Objects Migration**

When you encounter any critical errors while migrating pricing objects from BRM to PDC, run the `MigrateBRMPricing` utility with `-skip` parameter to skip critical errors and continue the migration process.

Typically, the migration process stops if critical errors are encountered. For pricing objects, you can run the `MigrateBRMPricing` utility with the `-skip` parameter to skip the critical errors encountered and continue the migration process. You can use the `-skip` parameter with `-pricing`, `-product`, `-discount`, and `-sponsorship` parameters. For example:

```
MigrateBRMPricing -product -skip
```

When you run the `MigrateBRMPricing` utility with `-skip` parameter, you are warned to ensure that you would want to skip objects and continue migration with errors. If you enter `Y` (yes), you are prompted to enter the encrypted password for the migration cross-reference database and the migration process is started after you provide the password. If you enter `N` (no), the migration process stops.

Running the `MigrateBRMPricing` utility with the `-skip` parameter:

1. Skips the following objects if there are critical errors encountered for these objects during the migration analysis phase: price selectors, Access Point Name (APN) selectors, price models, time models, usage scenario (USC) selectors, discount selectors, discount filters, and discount triggers.
2. Creates dummy objects for the following pricing objects while migrating the data: charges, discounts, discount selectors, charge selectors, and chargeshares.

You can find the dummy objects in the XML files in the `BRM_Integration_Pack_Home/apps/migration/xml/pdc` directory. After you complete the migration, you can search for the dummy objects in the PDC UI and fix the values.

```
Note: Dummy objects for charges and discounts are created with the pricing as 0 (zero) and the description as MIGRATION_DUMMY_OBJ.
```

For more information on the `MigrateBRMPricing` utility, see the discussion about `MigrateBRMPricing` in *PDC Pricing Migration Guide*.

**Fixing Compatibility "User Input Required" Errors**

The migration utility may require you to provide additional information prior to migrating configurations to the PDC database. The analysis reports list these objects marked as *User Input Required* with a description of the information that is required. You provide the input in an XML file.

To provide the input:

1. Make a copy of the predefined XML templates in the `BRM_Integration_Pack_Home/apps/xml` directory in the `BRM_Integration_Pack_Home/apps/xml/userinput` directory, where `BRM_Integration_Pack_Home` is the directory in which the PDC software is installed.
2. Update the XML file content with the required information.
3. Ensure that the XML input file conforms to the XML schema definition in the `BRM_Integration_Pack_Home/apps/xsd` directory.

See the following topics for more information:

- **Identifying Taxcode for the General Ledger ID Used in Offline Charging**
- **Mapping Batch Rating Engine Fields Used in Selectors to Corresponding ECE Fields**
- **Mapping Custom Rules To Usage Types in Selectors**
- **Identifying ECE Fields for RUM Fields in BRM**

### Identifying Taxcode for the General Ledger ID Used in Offline Charging

In PDC, general ledger IDs are mandatory in charge offers and charge selectors with the **Offline Usage** pricing profile. Each general ledger ID must be mapped to a tax code. ECE uses this mapping for offline charging. If you are migrating a charge offer or charge selector with the **Offline Usage** pricing profile, provide the XML file with the general ledger ID and tax code mapping as user input for migration.

#### Sample General Ledger ID and Taxcode Mapping XML File

If the general ledger IDs are 1031 and 1045 and the tax codes are NORM and TAX, the following is the XML that you would provide to map 1031 to NORM and 1045 to TAX:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<glidtaxcodemap>
  <glid value="1031">
    <taxcode>NORM</taxcode>
  </glid>
  <glid value="1045">
    <taxcode>TAX</taxcode>
  </glid>
</glidtaxcodemap>
```

### Mapping Batch Rating Engine Fields Used in Selectors to Corresponding ECE Fields

If you are migrating the batch rating engine pricing data to the PDC system that supports ECE, do the following:

1. Verify the source fields in the `BRM_Integration_Pack_Home/apps/xml/userinput/selectors field mapping XML file` (for example, `SelectorsFieldMappingUI.xml`).
2. Add or modify entries to map the fields from the batch rating engine container to the corresponding ECE fields.

#### Sample Selectors Field Mapping XML File

The following is an example of the XML file that you would provide to map batch rating engine fields to corresponding ECE fields:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<FieldNameMappings>
  <FieldNameMap>
    <SourceField>AnyEvent.ACCESS_POINT_NAME</SourceField>
    <TargetField>AnyEvent.ACCESS_POINT_NAME</TargetField>
    <FieldType>EventAttributeSpec</FieldType>
  </FieldNameMap>
</FieldNameMappings>
```
Fixing Errors Found in the Migration Phase

Mapping Custom Rules To Usage Types in Selectors
You need to map a custom rule to each usage type used in selectors.

To map the custom rule, you need to specify the custom rule as the target field in the selectors field mapping XML file. You can map the ready-to-use custom rules in the PDC_Home/apps/Samples/Examples/OOB_CRs.xml file (where PDC_Home is the directory in which the PDC software is installed) to the ready-to-use extended rating attributes (ERAs). If the custom rule uses a profile attribute specification, provide the profile attribute specification in the profile attribute specifications XML file (for example, ProfileAttrSpec.xml) as user input.

Sample Selectors Field Mapping XML File
The following is an example of the XML file that you would provide to map the Usage Type fields with the Custom Rule fields:

```xml
<?xml version="1.0" encoding='UTF-8' ?>
<FieldNameMappings>
  <FieldNameMap>
    <SourceField>AnyEvent.USAGE_TYPE</SourceField>
    <TargetField>Closed User Group Rule</TargetField>
    <FieldType>UsageTypeField.CUG</FieldType>
  </FieldNameMap>
  <FieldNameMap>
    <SourceField>AnyEvent.USAGE_TYPE</SourceField>
    <TargetField>Family Rule</TargetField>
    <FieldType>UsageTypeField.FF</FieldType>
  </FieldNameMap>
</FieldNameMappings>
```
Identifying ECE Fields for RUM Fields in BRM
If you are migrating the batch rating engine pricing data to the PDC system that supports ECE, do the following:

1. Verify the *fieldname* values in the `BRM_Integration_Pack_ Home/apps/xml/userinput/service-event mapping XML` file (for example, `BccServiceEventMapUserInput.xml`).

2. Specify the corresponding ECE field name for the RUM fields from the batch rating engine container.

Sample Service-Event Mapping XML File
The following is an example of the service-event mapping XML file that you would provide to use for the `TelcoGsmTelephony` service and `EventDelayedSessionTelcoGsm` event combination in ECE:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<attributeSpecMaps>
  <name>TelcoGsmData_ASM</name>
  <internalId>b8358331-f2ba-45c0-9d52-2efff62ca056</internalId>
  <priceListName>Default</priceListName>
  <eventRUMSpec>
    <name>EventDelayedSessionTelcoGsm_ERS</name>
    <priceListName>Default</priceListName>
    <eventSpecName>EventDelayedSessionTelcoGsm</eventSpecName>
    <rumSpec>
      <rumExpression>
        <eventFieldExpression>
          <fieldName>EventDelayedSessionTelcoGsm.DURATION</fieldName>
        </eventFieldExpression>
        <rumName>Duration_batch</rumName>
      </rumSpec>
      <rumSpec>
        <rumExpression>
          <eventFieldExpression>
            <fieldName>EventDelayedSessionTelcoGsm.WHOLESALE_AMOUNT</fieldName>
          </eventFieldExpression>
          <rumName>Passthrough</rumName>
        </rumSpec>
      </rumRUMSpec>
  </attributeSpecMaps>
</attributeSpecMaps>
```

Compatibility Errors Requiring Pricing Data Changes
The following are some common PDC configurations that require manual changes prior to migrating the data to the PDC system that supports ECE. These configurations are listed in the migration compatibility reports and the report also specifies exactly where the issues reside in your data. They are listed here for your reference and to provide more details on how to resolve them.

The following changes are required prior to migration:

- Events Split across Time Periods
- Quantity Ranges With Negative Values
Events Split across Time Periods
ECE supports only the Continuous splitting option for rating events that crosses multiple time periods. If you migrate a rate plan or charge with any other splitting option, the migration utility reports an error.

**Recommended Action**
When you migrate a PDC charge, ensure that the Quantity used to select steps field in the Advance section is set to Dependent On in the PDC UI or the Applicable Quantity Treatment element is set to Continuous in the XML file generated by the migration utility.

Quantity Ranges With Negative Values
ECE does not support quantity ranges with negative values. The migration utility reports an error if such values are migrated.

**Recommended Action**
When you migrate a PDC charge, or a discount, ensure that quantity ranges contain positive numbers.

Changes Required After Migration
The following changes are required after migration:
- Rate Plan Selector with Multiple RUMs
- Rate Plan Selector with Zone Models
- Non-Currency Resources Used as Counters
- BRM Rate Plans with Multiple Currencies

Rate Plan Selector with Multiple RUMs
In PDC, pricing using multiple RUMs is implemented with a single charge that contains separate charge trees for each RUM. In BRM, pricing using multiple RUMs is implemented with separate rate plans or rate plan selectors for each RUM.

For BRM rate plan selectors with multiple RUMs, the migration utility does not do a complete migration of the rate plan selectors. The rate plan selector is migrated to one charge selector with one rule mapped to a single charge and the charge will contain multiple charge trees, one for each RUM.

**Recommended Action**
After migration, re-configure the charge selector in PDC by defining rules where each rule results in a charge that contains a charge tree for each RUM.

Rate Plan Selector with Zone Models
In BRM, you can create rate plan selectors with zone models. But, ECE supports rate plan selectors only with value maps. Therefore, the migration utility does not migrate the zone model values in the rate plan selectors.

**Recommended Action**
After migration, add the zone model values for each charge and re-configure the charge selector in PDC by updating the rules.
Non-Currency Resources Used as Counters

When configuring a credit balance impact in PDC, only positive values are accepted in the PDC UI and the value is stored as a negative value, which is the convention for credits. Therefore, the value in the PDC UI has a different sign from the corresponding value in the database.

In Pricing Center, when a non-currency balance element is impacted with a positive value, as in a counter, it is migrated as follows:

- If Grantable flag is not checked in the balance impact, it is migrated to a debit in PDC.
- If Grantable flag is checked, it is migrated to a credit with a positive value in PDC. Because PDC reverses the sign of a credit amount, the migrated credit is initially displayed as a negative value, but when Save is clicked, the negative sign is removed, because negative values are not accepted in the UI. Consequently, the value in the database becomes a negative value. This changes the initial configuration and would likely result in pricing errors.

Recommended Action

After migration, change the balance element to a counter. If the balance element is tagged as a counter, the database value is not changed and PDC displays this balance impact as an increase of the counter using the actual amount stored in the database.

Example

Given the following balance impact in Pricing Center:

Impact: 'Dollars Spent'
Impact: 'Dollars Spent'
Id: 1500001
Amount: 0.40 per minute
Grantable: Yes

After migration, the balance impact in PDC is:

Credit: 'Dollars Spent'
Id: 1500001
Amount: 0.40 per minute

Without changing the balance element to a counter, if this balance impact is displayed in PDC and saved, it is stored as:

Credit: 'Dollars Spent'
Amount: -0.40 per minute
Id: 1500001

After changing the balance element to be a counter, if this balance impact is displayed in PDC, it is displayed and saved as:

Increase: 'Dollars Spent'
Id: 1500001
Amount: 0.40 per minute

Solution:

Update the BalanceElement table using the following SQL statement.

```
UPDATE BalanceElement
SET Counter=1
WHERE NumericCode=1500001;
```
BRM Rate Plans with Multiple Currencies

In BRM, a product can have multiple real-time rate plans, one rate plan per currency or one pipeline rate plan with multiple currencies. These product configurations are migrated in different ways.

BRM Real-Time Rate Plans

If you migrate a BRM product with multiple real-time rate plans per currency, it is migrated as one charge offer with one charge that has multiple branches, one for each currency charge. This is not displayed in the PDC application.

Recommended Action

After migration, if you want to change a price in the migrated charge offer, you can edit the XML for the charge offer, change prices for the appropriate currencies, and then import the charge offer without viewing it in a changeset in the PDC application. The changeset is stored in PDC and published to BRM.

BRM Pipeline Rate Plans

If you migrate a BRM product with a pipeline rate plan with multiple currencies, it is migrated as one charge offer with one charge that has multiple currencies. The currencies that are not specified for the rate plan are displayed as read-only in the PDC application.

Recommended Action

After migration, if you want, you can edit the charge and remove the currencies which are read-only and create duplicate charge offers with one charge per currency. For more information about configuring pricing for a charge, see the discussion about specifying charge details and configuring pricing in charges in *PDC Creating Product Offerings*.

Handling Migration Process Errors

Errors during the data migration can occur for various reasons, such as:

- Database connection problems
- Permission to write files
- The XML data does not comply with the XSD

If you encounter these types of errors, look in the migration utility’s log file, which is located in the log file directory specified in the *BRM_Integration_Pack_Home/apps/migration/MigrateConfiguration.xml* configuration file, for the specific details.

Resolving Java Heap Space Error

The migration utility is a Java process that runs on the Java Virtual Machine (JVM). While running the utility, you may receive "java.lang.OutOfMemoryError: Java heap space" error message on the WebLogic Managed Server. This occurs when the JVM runs out of heap space that it uses to run the utility.

The Java heap space size for the migration utility is set to 512 megabytes initial heap space and 4096 megabytes maximum heap space. In most cases, these default settings are sufficient. However, if you receive the Java heap space error, you may want to adjust the heap space sizes by adjusting the JVM parameters `-Xms<size>` and `-Xmx<size>`, where `-Xms<size>` specifies the initial heap space and `-Xmx<size>` specifies the maximum heap space.
To set the Java heap space size:

1. Open the `BRM_Integration_Pack_Home/apps/migration/MigrateBRMPricing` shell script in a text editor.

2. Search for the following line:

   ```
   ${JAVA_HOME}/bin/java -Xms512m -Xmx4096m -XX:CompileThreshold=8000 -XX:PermSize=128m -XX:MaxPermSize=1024m -cp ${clp} ${jvmopts} oracle.communications.brm.pdc.migration.Migrator $*
   ```

3. Change `512m` and `4096m` to appropriate heap sizes.

   For example:

   ```
   ${JAVA_HOME}/bin/java -Xms768m -Xmx4352m -XX:CompileThreshold=8000 -XX:PermSize=128m -XX:MaxPermSize=1024m -cp ${clp} ${jvmopts} oracle.communications.brm.pdc.migration.Migrator $*
   ```

4. Update the Java heap space settings on the WebLogic Server. See the Oracle WebLogic Server documentation for more information.

For more information about Java heap space sizing guidelines, refer to the JVM documentation.

**Restarting the Systems**

After the migration is complete, restart all the PDC and ECE system components. Restarting the systems clears the data that is cached in the system memory during the migration.

**Testing the Migrated Data**

After migration is complete, test the migrated data before moving the data to the production systems.

For example:

- Using the PDC application, verify that you are able to create and modify pricing and configuration objects without any errors.
- Generate some usage events in ECE, then perform rating and billing of the events using the pricing data from before the migration and after.
- Compare the rating and billing results and verify that the results are the same.

**Importing the Migrated Data to the Production Systems**

After you have successfully completed the migration of the configuration and pricing objects on the development system, you can import the changes to your production system.

**Important:** To import changes from the development system to the production system, it is required that the BRM or PDC database on the production system must have exactly the same pricing data configurations as in the BRM or PDC database on the development system prior to the original migration.
To import the data to the production system:

1. Ensure that the PDC production system is a new installation of PDC. The PDC database cannot have any existing data.

2. Ensure that the PDC production database contains exactly the same configuration and pricing objects as in the PDC database on the development system prior to the original migration.

3. Propagate any updates that you made to the PDC data in the development system during the original migration to the PDC production system.

4. Copy the client.jks from the target PDC production system into the source PDC system.

5. Update the BRM_Integration_Pack_Home/apps/bin/MigrateBRMPricing file with the keystore information for the target PDC production system where you want to import the changes:
   a. On the PDC development system, copy the BRM_Integration_Pack_Home/apps/bin/MigrateBRMPricing file and rename it Hostname_MigrateBRMPricing, where Hostname identifies the target PDC system.
   b. Edit the Hostname_MigrateBRMPricing file to reference the target production systems:
      a. Search for the jvmopts entry.
      b. Set the -Djavax.net.ssl.trustStore entry to the location of the client.jks file copied from Target server.
      c. Save and close the file.

6. Copy the BRM Integration Pack wallet from the target PDC production system into the source PDC system.

7. Update the migration utility’s configuration file with the BRM Integration Pack wallet and connection information for the source and target PDC production systems where you want to import the changes:
   a. On the PDC development system, copy the BRM_Integration_Pack_Home/apps/migration/MigrationConfiguration.xml file and rename it Hostname_MigrationConfiguration.xml, where Hostname identifies the target PDC system.
   b. Edit Hostname_MigrateConfiguration.xml to reference the target production systems. See “Configuring the Migration Utility” for more information:
      a. Search for walletConfiguration.
      b. Set the pdcWalletLocation entry to the location of the BRM Integration Pack wallet copied from the target PDC production system.
      c. Search for pricingServer.
      d. Edit the connectionInfo entry within the pricingServer element to reference the PDC production system.
      e. Search for the xrefDatabase element.

```xml
<xrefDatabase>
  <connectionInfo>
    <login>CrossRefUserName</login>
    <hostName>CrossRefHostName</hostName>
    <port>CrossRefPort</port>
    <serviceName>CrossRefServiceName</serviceName>
  </connectionInfo>
</xrefDatabase>
```
f. Edit the `connectionInfo` within the `xrefDatabase` element to reference the PDC transformation database.

g. Save and close the file.

8. Run the following command, which starts the import of the changes to the target PDC production system:

```
MigrateBRMPricing -retarget -properties Hostname_MigrateConfiguration.xml
```

If the retarget migration process stops, run the following command, which restarts the process:

```
MigrateBRMPricing -retarget -restart -properties Hostname_MigrateConfiguration.xml
```

When the retarget migration process is complete, a status report is generated in the report directory location specified in the `BRM_Integration_Pack_Home/apps/migration/MigrateConfiguration.xml` configuration file. The status report file name includes the target PDC system host name and port number. For example, `StatusReport_Retarget_hostname_portnumber.html`.

9. Remove the XML files generated during migration (Optionally, you can store them in a secure location if you need them for future use).

a. On the development system, open the `BRM_Integration_Pack_Home/apps/migration/MigrateConfiguration.xml` file.

b. Search for the `brmExtractedXML` element:

```
<brmExtractedXML>BRMExtractedXML</brmExtractedXML>
```

c. Remove the XML files in the directory location specified by `BRMExtractedXML`.

d. Search for the `brmXML` element:

```
<brmXML>BRMDataFile</brmXML>
```

e. Remove the XML files in the directory location specified by `BRMDataFile`.

f. Search for the `pdcXML` element:

```
<pdcXML>PDCDataFile</pdcXML>
```

g. Remove the XML files in the directory location specified by `PDCDataFile`.

### About Working with the PDC Pricing Data after Migration

After the BRM pricing data has been successfully migrated to PDC, you use the PDC application to modify the pricing data or create new pricing configurations. PDC transforms the data into the BRM pricing data model and then loads the data into the BRM database.

Modifications to the configuration objects which are mastered in BRM must be done in BRM and then synchronized with PDC by using `SyncPDC` utility.
Note: Ensure that /event/realtimeDiscount does not exist in BRM during migration.

See the discussion about synchronizing setup components in BRM PDC Creating Product Offerings for more information.
This chapter describes how to set up your environments and run the PDC migration utility to migrate the Oracle Communications Billing and Revenue Management (BRM) pricing data to a Oracle Communications Pricing Design Center (PDC) system that supports the real-time rating engine (RRE) and the batch rating engine (BRE).

Pricing data migration should be performed by experienced pricing administrators.

You should have a basic understanding of the following:

- Creating product offerings. See the discussion about creating product offerings in PDC Creating Product Offerings.
- Configuring setup components. See the discussion about configuring setup components in PDC Creating Product Offerings.

To use the migration utility, you must also understand the following:

- Database administration tasks
- Extensible Markup Language (XML) programming
- XML schema definition (XSD)

To minimize impact and risks to your production systems, Oracle recommends to perform the initial migration on a development test system and then import the migrated data to your production systems.

About Migrating BRM Pricing Data to PDC With RRE and BRE

If you are using Pricing Center to create and manage your price lists and using the real-time rating engine and the batch rating engine for usage rating, the pricing data is stored in the BRM database. If you want to use the PDC application to create and manage the price lists, you must first install PDC with RRE and BRE and migrate the pricing data from the BRM database into the PDC database. After you migrate the data, the pricing data is stored in the PDC database and used by the real-time and batch rating engines for usage rating.

You use the command-line utility, MigrateBRM Pricing, to migrate the BRM pricing data into the PDC database.

**Note:** The migration of branded data is not supported.
The migration utility migrates both real-time and batch pricing data. The BRM pricing data consists of configuration objects and pricing objects. Because the pricing objects reference configuration objects, you must migrate the configuration objects prior to migrating the pricing objects. To do this, you run the migration twice: first to migrate the configuration objects and then to migrate the pricing objects.

The configuration objects include data such as event objects, service objects, impact categories, and zone models.

The pricing objects consist of the following:

- Products and all objects referenced by a product, including rateplans, price model selectors, Access Point Name (APN) selectors, price models, time models, usage scenario (USC) groups.
- Discounts and all objects referenced by discounts, including discount models, discount model selectors, discount rules, triggers, and filters.
- Chargeshares and all objects referenced by a chargeshare.
- Subscriptions objects, including deals, plans, and plan lists.

If the pricing analysis reports contain a large number of messages, you can consider migrating pricing objects by category. If you decide to migrate the pricing objects by category, you must migrate them in the following order:

1. Products and all objects referenced in the products
2. Discounts and all the objects referenced in the discounts
3. Chargeshares
4. Subscription objects

If you migrate pricing objects by category, each migration process must be completed first before starting the next. For example, the products migration process must be completed, that means all products and related objects are analyzed, transformed, and available in the PDC database, before you start the discounts migration process and so on.

See "MigrateBRMPricing" for the utility’s syntax and parameters.

**Note:** The BRM term for an object sometimes differs from the PDC term for the same object. See the mapping BRM to PDC terminology table in *PDC Creating Product Offerings* for more information.

Migration of the BRM pricing data is performed in two phases:

1. **Data analysis:** In this phase, the migration utility extracts and analyzes the BRM pricing data. The analysis shows errors that need to be fixed before the data can be migrated.

2. **Data migration:** In this phase, the migration utility adjusts, transforms and migrates the BRM pricing data to the PDC database.

You must perform the migration process twice. First you analyze and migrate configuration objects, such as event objects, service objects, impact categories, and zone models. Then, you analyze and migrate pricing objects, such as products and discounts.
About the Data Analysis Phase for Migrating BRM Pricing Data to PDC With RRE and BRE

The data analysis phase is the first phase of the migration of the BRM pricing data. The following procedure describes how the BRM pricing data is processed in this phase:

1. The migration utility extracts the BRM pricing data from the BRM database and saves the data in XML files.
2. The migration utility analyzes the data to determine if the configurations are supported in PDC and whether any configurations need to be adjusted prior to the migration.
3. The migration utility generates analysis reports that provide the list of the data configurations that need to be adjusted by the utility during the migration process and any configurations that you would need to adjust manually.

You need to review the analysis reports and resolve any pricing data configuration errors in the BRM database. If you make updates to the pricing data in the BRM database, you must restart the migration process to extract the latest objects from the BRM database and perform the data analysis again.

When you restart the migration process, the analysis reports are also re-generated for the latest data.

Figure 10–1 shows the data analysis phase:

Figure 10–1  Data Analysis Phase for Migrating BRM Pricing Data to PDC With RRE and BRE

The migration utility can adjust most BRM pricing configurations. Adjustments to the BRM pricing data configurations may result in existing BRM objects being updated or new BRM objects being created. In some cases, the migration utility may require additional information from you before it can adjust the data. But, there are some configurations that it cannot adjust and for those cases, you may need to change the configurations manually before continuing with the migration.

The data analysis phase is complete when there are no data configurations errors in the analysis reports.

About the Data Migration Phase for Migrating BRM Pricing Data to PDC With RRE and BRE

The data migration phase is the second phase of the migration process. You must resume the migration process after the analysis phase is complete to start the data migration.

The following procedure describes how the data is processed in the data migration phase:
1. The migration utility adjusts the BRM pricing data in the XML files, if necessary, to enable transformation to the PDC data model.

2. The migration utility transforms the BRM pricing data to PDC components and saves them in XML files.

3. The migration utility loads the PDC components into the PDC database.

4. The migration utility loads any new and updated BRM objects (resulting from the data adjustments) into the BRM database.

5. The migration utility generates the migration report with information about the BRM objects that were successfully migrated to the PDC database.

6. The migration utility generates the process report with information about the BRM objects that were created or updated in the BRM database.

For more information about the migration utility reports, see "About Migration Reports".

Figure 10–2 shows the data migration phase:

Figure 10–2 Data Migration Phase for Migrating BRM Pricing Data to PDC With RRE and BRE

For more information on setting up and migrating the BRM pricing data to PDC with the real-time rating engine and the batch rating engine, see "Performing Migration of BRM Pricing Data to PDC With RRE and BRE".

Performing Migration of BRM Pricing Data to PDC With RRE and BRE

The following steps guide you to set up and run migration on the development system and then import the changes to the production databases:

1. Create a backup of the BRM production database.

See “Creating a BackUp of Your BRM Database".
2. Setup the development system to run the initial migration of the pricing data. See "Setting Up the BRM Development System" for more information.

3. Migrate the BRM configuration and pricing objects. See "Migrating the BRM Pricing Data to PDC With RRE and BRE".

4. Restart the BRM and PDC systems. See "Restarting the Systems".

5. Test the migrated data. See "Testing the Migrated Data".

6. Import the migrated data to the production systems. See "Importing the Migrated Data to the Production Systems".

Creating a BackUp of Your BRM Database

Oracle recommends that you perform a full backup of your BRM production database prior to running the migration.

Performing a full backup allows you to restore your production database to its original state in the event of any data corruption or data loss. Back up both the database definition and all the database contents.

Use the backup to restore the database on a test system to ensure the backup is valid. See your database software documentation for more information on performing full database backups.

Setting Up the BRM Development System

Setting up the BRM development system includes setting up a BRM system with BRM pricing data and installing a PDC system.

Perform the following tasks to setup your development system:

1. Review the system requirements for installing the PDC and BRM systems. See the discussion about Pricing Design Center system requirements in PDC Installation Guide.

2. Perform the following pre-installation tasks for PDC:
   - Installing and configuring the Oracle Database
   - Installing and configuring Oracle WebLogic Server
   - Installing and configuring BRM
   See the discussion about Pricing Design Center pre-Installation tasks in PDC Installation Guide.

3. Install the complete PDC software.
4. Copy the BRM pricing data from the production database to the development system.

You can copy the BRM pricing data in one of two ways:

- Use the BRM load utilities, `loadpricelist` and `LoadIfwConfig`, to export the pricing data from your BRM production database and load it into the BRM development database.

See the discussion about pricing utilities in *PDC Creating Product Offerings* for information about running these utilities.

- Restore the database from the BRM database backup.

See your database software documentation for more information about database restore.

5. Ensure `BRM_Integration_Pack_Home/apps/bin` is in your PATH environment variable, where `BRM_Integration_Pack_Home` is the directory in which you installed BRM Integration Pack.

6. (Optional) Configure the `MigrateBRMPricing` utility. See "Configuring the Migration Utility" for more information.

7. Ensure all users of the migration utility are added to the `Migration Admin` group.

The users of the migration utility must belong to the `Migration Admin` group, which is created during the PDC installation. You can add additional users to this group by using the WebLogic Administration Console.

8. Ensure that the BRM `loadpricelist` and `LoadIfwConfig` utilities are configured correctly and can connect to the BRM database.

The migration utility uses the BRM load utilities to extract the pricing data from the BRM database to XML files.
See the discussion about pricing utilities in *PDC Creating Product Offerings* for more information.

9. (Optional) If your BRM data consists of pipeline services that are mapped to multiple events in the Pipeline Manager IFW_REF_MAP database table, then you must add REF_PARAM combination key to the IFW_REF_MAP entry in IFW_Home/tools/XmlLoader/LoadIfwConfig.xsd file.

   <xs:element name="IFW_REF_MAP" type="TableType_IFW_REF_MAP" minOccurs="0" maxOccurs="unbounded">
     <xs:key name="IFW_REF_MAP_PrimaryKey_1">
       <xs:selector xpath="IFW_REF_MAP"/>
       <xs:field xpath="@ID"/>
       <xs:field xpath="@REF_OBJ"/>
       <xs:field xpath="@REF_PARAM"/>
     </xs:key>
   </xs:element>

10. Obtain the Java Keystore password.

### Configuring the Migration Utility

The migration configuration file contains the information that you provided during PDC installation. You can edit this file if you want to change the information that you provided.

To configure the migration utility:

1. Make a copy of the `BRM_Integration_Pack_Home/apps/migration/MigrateConfiguration.xml` file, where `BRM_Integration_Pack_Home` is the directory in which you installed BRM Integration Pack.

2. Open the copy in a text editor.

3. Edit the file based on your requirements.

   *Table 10–1* lists the elements in `MigrateConfiguration.xml` and the syntax and description for each element.
### Table 10–1  Elements in the MigrateConfiguration.xml File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>walletConfiguration</td>
<td><code>&lt;walletConfiguration&gt;</code>&lt;pdcWalletLocation&gt;BRMIntegrationPackWallet&lt;/pdcWalletLocation&gt;<code>walletConfiguration&gt;</code></td>
<td>Specifies the Oracle wallet that stores the sensitive information, such as database passwords, where <code>BRMIntegrationPackWallet</code> is the complete path to the BRM Integration Pack wallet.</td>
</tr>
</tbody>
</table>
| xrefDatabase      | `<xrefDatabase>`<connectionInfo><login>CrossRefUserName</login><hostName>CrossRefHostName</hostName><port>CrossRefPort</port><serviceName>CrossRefServiceName</serviceName></connectionInfo>`xrefDatabase>` | Contains the details about the transformation cross-reference database, where:  
  - `CrossRefUserName` specifies the cross-reference database user name  
  - `CrossRefHostName` specifies the IP address or the host name of the machine on which the cross-reference database is configured  
  - `CrossRefPort` specifies the port number assigned to the cross-reference database  
  - `CrossRefServiceName` specifies the name of the cross-reference database service |
| migrationDatabase | `<migrationDatabase>`<connectionInfo><login>MigrationUserName</login><hostName>MigrationHostName</hostName><port>MigrationPort</port><serviceName>MigrationServiceName</serviceName></connectionInfo>`migrationDatabase>` | Contains the details about the migration cross-reference database, where:  
  - `MigrationUserName` specifies the cross-reference database user name  
  - `MigrationHostName` specifies the IP address or the host name of the machine on which the cross-reference database is configured  
  - `MigrationPort` specifies the port number assigned to the cross-reference database  
  - `MigrationServiceName` specifies the name of the cross-reference database service  
  By default, the connection information in the `<migrateDatabase>` elements is same as the connection information in the `<xrefDatabase>` elements. However, during BRM Integration Pack installation, the Installer allows you specify a migration cross-reference database that is different from the transformation cross-reference database. In that case, the connection information in the `<migrateDatabase>` elements and the `<xrefDatabase>` elements may differ. |
| migrationSource   | `<migrationSource>`<deploymentType>MigrationSourceType</deploymentType><pricingServer>SourcePDCPricingServer</pricingServer><brmConfiguration>SourceBRMConfiguration</brmConfiguration>`migrationSource>` | Contains the details about the migration source system.  
  - `<deploymentType>` element specifies the migration source, where `MigrationSourceType` is `LEGACY_BRM` to specify that the source is a BRM system that uses real-time and batch rating engines and BRM Pricing Center.  
  - `<pricingServer>` element is not applicable for `LEGACY_BRM`.  
  - `<brmConfiguration>` element contains the source BRM system configuration details. |
Setting Up the BRM Development System

Table 10–1  (Cont.) Elements in the MigrateConfiguration.xml File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pricingServer</td>
<td>&lt;migrationSource&gt; &lt;pricingServer&gt; &lt;connectionInfo&gt;</td>
<td>Contains the source PDC server information, where:</td>
</tr>
<tr>
<td></td>
<td>&lt;hostName&gt;SourcePricingServerHostName&lt;/hostName&gt;</td>
<td>■ SourcePricingServerHostName specifies the IP address or the host name of</td>
</tr>
<tr>
<td></td>
<td>&lt;port&gt;SourcePricingServerPort</td>
<td>the machine on which the source PDC server is deployed</td>
</tr>
<tr>
<td></td>
<td>&lt;adminUser&gt;SourceAdminUserName&lt;/adminUser&gt;</td>
<td>■ SourcePricingServerPort specifies the port number of the domain on which</td>
</tr>
<tr>
<td></td>
<td>&lt;pdcUser&gt;SourcePDCUser&lt;/pdcUser&gt;</td>
<td>the source PDC server is deployed</td>
</tr>
<tr>
<td></td>
<td>&lt;pdcSSL&gt;SourceSSLOption&lt;/pdcSSL&gt;</td>
<td>■ SourceAdminUserName specifies the user name of the source PDC server</td>
</tr>
<tr>
<td></td>
<td>&lt;/connectionInfo&gt;</td>
<td>administrator</td>
</tr>
<tr>
<td></td>
<td>&lt;/pricingServer&gt;</td>
<td>■ SourcePDCUser specifies the user name of the source PDC system user</td>
</tr>
<tr>
<td></td>
<td>&lt;/migrationSource&gt;</td>
<td>■ SourceSSLOption specifies whether the source PDC server supports SSL.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>enabled. Specifies that PDC supports SSL. If SSL is enabled, MigrateBRMPricing uses the t3s://Host:Port URL to access PDC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>disabled. Specifies that PDC does not support SSL. If SSL is disabled,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MigrateBRMPricing uses the t3://Host:Port URL to access PDC</td>
</tr>
</tbody>
</table>
### Setting Up the BRM Development System

#### Elements in the MigrateConfiguration.xml File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>brmConfiguration</td>
<td><code>&lt;migrationSource&gt;</code>&lt;brmConfiguration&gt;&lt;brand&gt;BrandOption&lt;/brand&gt;&lt;brmConfiguration&gt;&lt;fieldSelection&gt;EventFieldInfo&lt;/fieldSelection&gt;&lt;breConfig&gt;&lt;containerDesc&gt;&lt;param&gt;&lt;paramname&gt;EDRField&lt;/paramname&gt;&lt;paramvalue&gt;EDRFieldValue&lt;/param&gt;&lt;param&gt;&lt;paramname&gt;BREEvent&lt;/paramname&gt;&lt;paramvalue&gt;ExtensionBlockName&lt;/param&gt;&lt;/containerDesc&gt;&lt;eventExtension&gt;&lt;param&gt;&lt;paramname&gt;/event/delayed/session/telco/gsm&lt;/paramname&gt;&lt;paramvalue&gt;ASS_GSMW_EXT&lt;/param&gt;&lt;/param&gt;&lt;/eventExtension&gt;&lt;serviceExtension&gt;&lt;param&gt;&lt;paramname&gt;/service/telco/gprs&lt;/paramname&gt;&lt;paramvalue&gt;ASS_GPRS_SERV_EXT&lt;/param&gt;&lt;/param&gt;&lt;/serviceExtension&gt;&lt;defaultValue&gt;&lt;param&gt;&lt;paramname&gt;SERVICE_CLASS&lt;/paramname&gt;&lt;paramvalue&gt;DefaultServiceClass&lt;/paramvalue&gt;&lt;/param&gt;&lt;/defaultValue&gt;&lt;breConfig&gt;&lt;/breConfig&gt;&lt;/brmConfiguration&gt;</td>
<td>Contain the details about the source BRM system configuration.</td>
</tr>
</tbody>
</table>
|                  | `</migrationSource>`                                                     | **<brand>** element specifies whether the source BRM system supports branding, where `BrandOption` is:  
  - **enabled.** to specify that the source BRM system supports branding  
  - **disabled.** to specify that the source BRM system does not support branding  
  
  **<skipBREMigra** element specifies to skip migration of pipeline configuration data, where `SkipOption` is either true or false.  

  **<fieldSelection>** element contains the information about the target rating engine and the BRM event fields provided as input to the target rating engine for usage rating.  

  **<breConfig>** element contains the batch rating engine configurations.  

  **<containerDesc>** element specifies the EDRC fields that are migrated for events.  

  In the following example, all the EDRC_FIELDS mapped to `ALL_RATE` are migrated for the events.  

  ```xml  
  <containerDesc>  
  <param><paramname>EDR_FIELD_NAME</paramname><paramvalue>ALL_RATE</paramvalue></param>  
  </containerDesc>  
  ```  

  **<eventExtension>** element specifies to migrate the fields of the specified extension block for the batch rating event.  

  Example:  

  ```xml  
  <eventExtension>  
  <param><paramname>/event/delayed/session/telco/gsm</paramname><paramvalue>ASS_GSMW_EXT</paramvalue></param>  
  </eventExtension>  
  ```  

  **<serviceExtension>** element specifies to migrate the fields of the specified extension block for the batch rating engine service.  

  Example:  

  ```xml  
  <serviceExtension>  
  <param><paramname>/service/telco/gprs</paramname><paramvalue>ASS_GPRS_SERV_EXT</paramvalue></param>  
  </serviceExtension>  
  ```  

  **<defaultValue>** element specifies the default service class, where `DefaultServiceClass` is the default service class name.  

<table>
<thead>
<tr>
<th></th>
<th><code>&lt;/migrationSource&gt;</code></th>
<th></th>
</tr>
</thead>
</table>

---

**Table 10–1 (Cont.) Elements in the MigrateConfiguration.xml File**
### Table 10–1 (Cont.) Elements in the MigrateConfiguration.xml File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fieldSelection</td>
<td><code>&lt;fieldSelection&gt;</code>&lt;br&gt;<code>&lt;targetEngine&gt;TargetRatingEngine&lt;/targetEngine&gt;</code>&lt;br&gt;<code>&lt;eventFields&gt;</code>&lt;br&gt;<code>&lt;eventName&gt;EventName&lt;/eventName&gt;</code>&lt;br&gt;<code>&lt;fullyQualifiedName&gt;EventFieldName&lt;/fullyQualifiedName&gt;</code>&lt;br&gt;<code>&lt;/eventFields&gt;</code>&lt;br&gt;<code>&lt;/fieldSelection&gt;</code>&lt;br&gt;</td>
<td>Contains the information about the target rating engine and the BRM event fields provided as input to the target rating engine for usage rating.&lt;br&gt;<code>&lt;targetEngine&gt;</code> element specifies the target rating engine used for usage rating, where:&lt;br&gt;  ■ <strong>Convergent Charging</strong>. Specifies that ECE is used for usage rating.&lt;br&gt;  ■ <strong>Realtime Charging</strong>. Specifies that the real-time rating engine is used for usage rating.&lt;br&gt;  ■ <strong>Batch Charging</strong>. Specifies that the batch rating engine is used for usage rating.&lt;br&gt;<code>&lt;eventFields&gt;</code> element specifies the fields that are provided as input to <code>&lt;targetEngine&gt;</code>, where:&lt;br&gt;  ■ <strong>eventName</strong> specifies the class name of the BRM event; for example, <code>/event</code>.&lt;br&gt;  ■ <strong>fullyQualifiedName</strong> specifies the fully qualified name of the BRM event field that is provided as input to <code>&lt;targetEngine&gt;</code>; for example, <code>PIN_FLD_NAME</code>.&lt;br&gt;Note: Each <code>&lt;fieldSelection&gt;</code> element can have multiple <code>&lt;eventFields&gt;</code> elements but only one <code>&lt;targetEngine&gt;</code> element. To support multiple target rating engines, add the <code>&lt;fieldSelection&gt;</code> element for each target rating engine.&lt;br&gt;Similarly, each <code>&lt;eventFields&gt;</code> element can have multiple <code>&lt;fullyQualifiedName&gt;</code> elements but only one <code>&lt;eventName&gt;</code> element. To rate multiple events, add the <code>&lt;eventFields&gt;</code> element for each event.</td>
</tr>
<tr>
<td>migrationTarget</td>
<td><code>&lt;migrationTarget&gt;</code>&lt;br&gt;<code>&lt;deploymentType&gt;MigrationTargettype&lt;/deploymentType&gt;</code>&lt;br&gt;<code>&lt;pricingServer&gt;TargetPDCPricingServer&lt;/pricingServer&gt;</code>&lt;br&gt;<code>&lt;brmConfiguration&gt;TargetBRMConfiguration&lt;/brmConfiguration&gt;</code>&lt;br&gt;<code>&lt;/migrationTarget&gt;</code>&lt;br&gt;</td>
<td>Contains the details about the migration target system.&lt;br&gt;  ■ <code>&lt;deploymentType&gt;</code> element specifies the migration target, where <code>MigrationTargettype</code> is <strong>PDC_WITH_ONLINE_BATCH_USAGE</strong> to specify that the target is a BRM system that uses real-time and batch rating engines and PDC.&lt;br&gt;  ■ <code>&lt;pricingServer&gt;</code> element contains the target PDC server information&lt;br&gt;  ■ <code>&lt;brmConfiguration&gt;</code> element contains the target BRM system configuration details.</td>
</tr>
<tr>
<td>pricingServer</td>
<td><code>&lt;migrationTarget&gt;</code>&lt;br&gt;<code>&lt;pricingServer&gt;</code>&lt;br&gt;<code>&lt;connectionInfo&gt;</code>&lt;br&gt;<code>&lt;hostName&gt;TargetPricingServerHostName&lt;/hostName&gt;</code>&lt;br&gt;<code>&lt;port&gt;TargetPricingServerPort&lt;/port&gt;</code>&lt;br&gt;<code>&lt;adminUser&gt;TargetAdminUserName&lt;/adminUser&gt;</code>&lt;br&gt;<code>&lt;pdcUser&gt;TargetPDCUser&lt;/pdcUser&gt;</code>&lt;br&gt;<code>&lt;pdcSSL&gt;TargetSSLOption&lt;/pdcSSL&gt;</code>&lt;br&gt;<code>&lt;/connectionInfo&gt;</code>&lt;br&gt;<code>&lt;/pricingServer&gt;</code>&lt;br&gt;<code>&lt;/migrationTarget&gt;</code>&lt;br&gt;</td>
<td>Contains the target PDC server information, where:&lt;br&gt;  ■ <strong>TargetPricingServerHostName</strong> specifies the IP address or the host name of the machine on which the target PDC server is deployed&lt;br&gt;  ■ <strong>TargetPricingServerPort</strong> specifies the port number of the domain on which the target PDC server is deployed&lt;br&gt;  ■ <strong>TargetAdminUserName</strong> specifies the user name of the target PDC server administrator&lt;br&gt;  ■ <strong>TargetPDCUser</strong> specifies the user name of the target PDC system user&lt;br&gt;  ■ <strong>TargetSSLOption</strong> specifies whether the target PDC server supports SSL.&lt;br&gt;  ■ <strong>enabled</strong>. Specifies that PDC supports SSL. If SSL is enabled, <strong>MigrateBRMPricing</strong> uses the <code>t3d/Host:Port</code> URL to access PDC.&lt;br&gt;  ■ <strong>disabled</strong>. Specifies that PDC does not support SSL. If SSL is disabled, <strong>MigrateBRMPricing</strong> uses the <code>t3d/Host:Port</code> URL to access PDC.</td>
</tr>
</tbody>
</table>
4. Save and close the file.

### Migrating the BRM Pricing Data to PDC With RRE and BRE

When you migrate pricing data from the BRM database to the PDC database, you must migrate the configuration objects and then migrate the pricing objects.

To migrate the pricing data from the BRM database to the PDC database:

1. Migrate the configuration objects. See "Migrating Configuration Objects".
2. Migrate the pricing objects. See "Migrating Pricing Objects".

---

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
</table>
| brmConfiguration | `<brmConfiguration>`  
|                  | `<loadConfigDir>LoadConfigData</loadConfigDir>`  
|                  | `<loadPriceListDir>LoadPriceData</loadPriceListDir>`  
|                  | `<loadIfwConfigDir>LoadPipelineConfigData</loadIfwConfigDir>`  
|                  | `/brmConfiguration`  
|                  | `/migrationTarget>`  
|                  | `<migrationTarget>`  
|                  | ` Contain the details about the target BRM system configuration, where:  
|                  | - `LoadConfigPath` specifies the path to the directory from where the `load_config` utility is run.  
|                  | - `LoadPriceListPath` specifies the path to the directory from where the `load_price_list` utility is run.  
|                  | - `LoadPipelineConfigPath` specifies the path to the directory from where the `LoadIfwConfig` utility is run.  
| logFile          | `<logFile>`  
|                  | `<logFile>`  
|                  | `MigrationLogFileLocation`  
|                  | `/logFile>`  
|                  | ` Specifies the directory that stores MigrateBRMPricing log files, where MigrationLogFileLocation is the complete path and the name of the log file.  
| reportFile       | `<reportFile>`  
|                  | `<reportFile>`  
|                  | `ReportFileLocation`  
|                  | `/reportFile>`  
|                  | ` Specifies the directory where MigrateBRMPricing stores the reports generated during the migration process, where ReportFileLocation is the complete path to the directory.  
| brmExtractedXML  | `<brmExtractedXML>`  
|                  | `<brmExtractedXML>`  
|                  | `BRMExtractedXML`  
|                  | `/brmExtractedXML>`  
|                  | ` Specifies the directory where MigrateBRMPricing stores the XML files containing the pricing data extracted from the BRM database, where BRMExtractedXML is the complete path to the directory.  
| pdcExtractedXML  | `<pdcExtractedXML>`  
|                  | `<pdcExtractedXML>`  
|                  | `PDCExtractedXML`  
|                  | `/pdcExtractedXML>`  
|                  | ` Specifies the directory where MigrateBRMPricing stores the XML files containing the pricing data extracted from the PDC database, where PDCExtractedXML is the complete path to the directory.  
| brmXML           | `<brmXML>`  
|                  | `<brmXML>`  
|                  | `BRMDataFile`  
|                  | `/brmXML>`  
|                  | ` Specifies the directory where MigrateBRMPricing stores the XML files containing the pricing data that needs to be updated or created in the BRM database, where BRMDataFile is the complete path to the directory.  
| pdcXML           | `<pdcXML>`  
|                  | `<pdcXML>`  
|                  | `PDCDataFile`  
|                  | `/pdcXML>`  
|                  | ` Specifies the directory where MigrateBRMPricing creates the XML files containing the extracted BRM data in PDC format, where PDCDataFile is the complete path to the directory.  
| userInputXML     | `<userInputXML>`  
|                  | `<userInputXML>`  
|                  | `UserInputXML`  
|                  | `/userInputXML>`  
|                  | ` Specifies the directory that contains the XML user input files that you provide for data adjustments, where UserInputXML is the complete path to the directory.  
| xrefSQL          | `<xrefSQL>`  
|                  | `<xrefSQL>`  
|                  | `XrefData`  
|                  | `/xrefSQL>`  
|                  | ` Specifies the directory where MigrateBRMPricing creates the files containing the migration and transformation cross-reference data in sql format, where XrefData is the complete path to the directory.  

---

Table 10–1 (Cont.) Elements in the MigrateConfiguration.xml File
Migrating Configuration Objects

To migrate the configuration objects:

1. Ensure that the BRM and PDC databases are running.
2. Go to the BRM_Integration_Pack_Home/apps/migration directory.
3. Run the following command, which starts the data analysis phase:
   ```
   MigrateBRMPricing -config -analyze
   ```
4. Enter the BRM Integration Pack wallet password when prompted.
5. Review the configuration analysis report and fix any data configuration errors that are reported (see "Fixing Errors Found in the Analysis Phase").
6. In step 5, if you made changes to the data in the BRM database, run the following command, which restarts the data analysis phase:
   ```
   MigrateBRMPricing -config -analyze -restart
   ```
7. Repeat steps 5 and 6 until there are no errors in the configuration analysis report.
8. Run the following command, which starts the data migration phase:
   ```
   MigrateBRMPricing -resume
   ```
9. Review the migration report and verify that all the configuration objects were migrated to the PDC database successfully.
10. Using the target PDC application, verify that you are able to view, create, and modify configuration objects without any errors.

Migrating Pricing Objects

To migrate pricing objects, do one of the following:

- (Recommended) Migrate all pricing objects in one migration process. See "Migrating All Pricing Objects".
- Migrate pricing objects by category. See "Migrating Pricing Objects by Category".

Migrating All Pricing Objects

To migrate all pricing objects:

1. In the PDC application, ensure that the configuration objects are available.
2. Ensure that the BRM and PDC databases are running.
3. Go to the BRM_Integration_Pack_Home/apps/migration directory.
4. Run the following command, which starts the data analysis phase for all pricing objects:
   ```
   MigrateBRMPricing -pricing -analyze
   ```

**Note:** MigrateBRMPricing does not proceed with the transformation and migration of the configuration objects to the PDC database if there are any critical or user input errors in the configuration analysis report.
Enter the BRM Integration Pack wallet password when prompted.

Review the pricing analysis reports (product, discount, sponsorship and subscription analysis reports) and fix any data configuration errors that are reported (see "Fixing Errors Found in the Analysis Phase").

Do one of the following:
- If you fixed any configuration object errors reported in the pricing analysis reports, restart the migration by running the configuration migration process and then rerunning the pricing migration process. See "Migrating the BRM Pricing Data to PDC With RRE and BRE".
- If you fixed pricing objects errors (only pricing objects errors reported in the pricing analysis report), restart the pricing data analysis phase by running the following command:

  `MigrateBRMPricing -pricing -analyze -restart`

Repeat steps 6 and 7 until there are no errors in the pricing analysis reports.

Run the following command, which starts the data migration phase:

`MigrateBRMPricing -resume`

---

**Note:** `MigrateBRMPricing` does not proceed with the transformation and migration of the pricing objects to the PDC database if there are any critical or user input errors in the pricing analysis reports.

Review the migration report and verify that all pricing objects were migrated to PDC successfully.

Make corrections to PDC data, if needed. See "Changes Required After Migration".

**Migrating Pricing Objects by Category**

When you migrate pricing objects by category, the migration of all the objects in each category must be completed before starting the next.

To migrate pricing objects by category:

1. In the PDC application, ensure that the configuration objects are available.
2. Ensure that the BRM and PDC databases are running.
3. Go to the `BRM_Integration_Pack_Home/apps/migration` directory.
4. Run the following command, which starts the data analysis phase for products and all objects referenced by a product:

   `MigrateBRMPricing -product -analyze`

5. Enter the BRM Integration Pack wallet password when prompted.
6. Review the product analysis report and fix any data configuration errors that are reported (see "Fixing Errors Found in the Analysis Phase").
7. Do one of the following:
   - If you fixed any configuration object errors reported in the product analysis report, restart the migration by running the configuration migration process and then rerunning the pricing migration process. See "Migrating the BRM Pricing Data to PDC With RRE and BRE".
Migrating the BRM Pricing Data to PDC With RRE and BRE

If you fixed pricing objects errors (only pricing objects errors reported in the product analysis report), restart the data analysis phase for products by running the following command:

```
MigrateBRMPricing -product -analyze -restart
```

8. Repeat steps 6 and 7 until there are no errors in the product analysis report.

9. Run the following command, which starts the data migration phase for products:

```
MigrateBRMPricing -resume
```

10. Review the migration report and verify that the products and objects referenced by a product were migrated to PDC successfully.

11. Make corrections to PDC data, if needed. See "Changes Required After Migration".

12. Run the following command, which starts the data analysis phase for discounts and all objects referenced by a discount:

```
MigrateBRMPricing -discount -analyze
```

13. Enter the BRM Integration Pack wallet password when prompted.

14. Review the discount analysis report and fix any data configuration errors that are reported.

15. Do one of the following:

   - If you fixed any configuration object errors reported in the discount analysis report, restart the migration by running the configuration migration process and then rerunning the pricing migration process. See "Migrating the BRM Pricing Data to PDC With RRE and BRE".

   - If you fixed pricing objects errors (only pricing objects errors reported in the discount analysis report), restart the data analysis phase for discounts by running the following command:

     ``
     MigrateBRMPricing -discount -analyze -restart
     ```

16. Repeat steps 14 and 15 until there are no errors in the discount analysis report.

17. Run the following command, which starts the data migration phase for discounts:

```
MigrateBRMPricing -resume
```

18. Review the migration report and verify that the discounts and all objects referenced by a discount were migrated to PDC successfully.

19. Make corrections to PDC data, if needed. See "Changes Required After Migration".

20. Run the following command, which starts the data analysis phase for chargeshares and all objects referenced by a chargeshare:

```
MigrateBRMPricing -sponsorship -analyze
```

**Note:** MigrateBRMPricing does not proceed with the transformation and migration of the pricing objects to the PDC database if there are any critical or user input errors in the analysis reports.
21. Enter the BRM Integration Pack wallet password when prompted.

22. Review the sponsorship analysis report and fix any data configuration errors that are reported.

23. Do one of the following:
   ■ If you fixed any configuration object errors reported in the sponsorship analysis report, restart the migration by running the configuration migration process and then rerunning the pricing migration process. See "Migrating the BRM Pricing Data to PDC With RRE and BRE".
   ■ If you fixed pricing objects errors (only pricing objects errors reported in the sponsorship analysis report), restart the data analysis phase for chargeshares by running the following command:

     `MigrateBRMPricing -sponsorship -analyze -restart`

24. Repeat steps 22 and 23 until there are no errors in the sponsorship analysis report.

25. Run the following command, which starts the data migration phase for chargeshares:

     `MigrateBRMPricing -resume`

26. Review the migration report and verify that the chargeshares and all objects referenced by a chargeshare were migrated to PDC successfully.

27. Make corrections to PDC data, if needed. See "Changes Required After Migration".

28. Run the following command, which starts the data analysis phase for subscription objects:

     `MigrateBRMPricing -subscription -analyze`

29. Enter the BRM Integration Pack wallet password when prompted.

30. Review the subscription analysis report and fix any data configuration errors that are reported.

31. Do one of the following:
   ■ If you fixed any configuration object errors reported in the subscription analysis report, restart the migration by running the configuration migration process and then rerunning the pricing migration process. See "Migrating the BRM Pricing Data to PDC With RRE and BRE".
   ■ If you fixed pricing objects errors (only pricing objects errors reported in the subscription analysis report), restart the data analysis phase for subscription objects by running the following command:

     `MigrateBRMPricing -subscription -analyze -restart`

32. Repeat steps 30 and 31 until there are no errors in the subscription analysis report.

33. Run the following command, which starts the data migration phase for subscription objects:

     `MigrateBRMPricing -resume`

34. Review the migration report and verify that all subscription objects were migrated to PDC successfully.

35. Make corrections to PDC data, if needed. See "Changes Required After Migration".
Fixing Errors Found in the Analysis Phase

The following sections describe how to provide required user input and how to fix some of the common critical errors reported in the migration analysis reports.

Fixing Analysis "User Input Required" Errors

The migration utility may require you to provide additional information prior to migrating resources and discount and chargeshare configurations. The analysis reports list these objects marked as User Input Required with a description of the information that is required. You provide the input in an XML file.

To provide the input:

1. Make a copy of the predefined XML templates in the BRM_Integration_Pack_Home/apps/xml directory in the BRM_Integration_Pack_Home/apps/xml/userinput directory, where BRM_Integration_Pack_Home is the directory in which the PDC software is installed.

2. Update the XML file content with the required information.

3. Ensure that the XML input file conforms to the XML schema definition in the BRM_Integration_Pack_Home/apps/xsd directory.

Identifying Resource Id for Non-Currency Resources Used in Batch Rating

If you have a non-currency resource that is used in batch rating, you need to map the resource name to the resource Id defined in the BRM server so that the resource can be mapped to the BEID definition.

The following is the XML that you would provide to map the resource LOYALTY to resource Id 1000019.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Resource>
  <BalanceElement type="0">
    <BREResourceName>LOYALTY</BREResourceName>
    <BEIDCode>LOY</BEIDCode>
    <ResId>1000019</ResId>
  </BalanceElement>
</Resource>
```

Providing Values List for Expressions in Discount and Chargeshare Configurations

If you use an expression as a filter criteria in the discount and chargeshare detail, you need to list the values in the expression individually.

Example 1

If you use the expression Peak* with Peak1, Peak2, and Peak3 for time model and 21* with 2155557 and 2155558 for resource Id in the discount detail, you need to provide these values individually. Following is an example of the XML file that you would provide:

```xml
<Filters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="oracle\communications\brm\pdc\migration\pricing\impl\bre\userinputxml\BreFilter_UserInput.xsd">
  <Filter master="M1" rank="1">
    <TimeModel>
      <name>Peak1</name>
      <name>Peak2</name>
      <name>Peak3</name>
    </TimeModel>
  </Filter>
</Filters>
```
Alternatively, you can correct the expression in the discount detail itself. If you change the discount detail configuration, you need to restart the migration utility.

Example 2

If you specify an expression in the rate plan field, such as !(Rate1 | Rate2 | Rate3), you need to provide the XML that specifies the rate plans that should apply rather than those that should be excluded. This is because the NOT operator is not supported in PDC.

Analysis Errors Requiring Pricing Data Changes

The following are some common BRM configurations that require manual changes prior to migrating the data to the PDC data model. These configurations are listed in the migration analysis reports and the report also specifies exactly where the issues reside in your data. They are listed here for your reference and to provide more details on how to resolve them.

The following changes are required prior to migration:

- Multiple Impact Categories in a Rate Plan
- Derived Impact Category Used in Zone Model
- Zone Items Differentiated Only by Service Code
- Extended Service Class Event Mapping
- Service Contains Deal Mapped to Parent Service
- Using Overlapping Date Ranges to Sequentially Impact Multiple Resources
- Quantity Brackets in a Fold
- BRM Object Names Containing Special Characters
- Custom Fields

Multiple Impact Categories in a Rate Plan

In PDC, the balance impact in charge does not contain impact categories. If you migrate a BRM rate plan selector with multiple rules which result in the same rate plan but with different impact categories, the migration utility reports an error. If you migrate a rate plan selector with multiple rules that result in the same rate plan with the same impact category, this rate plan selector is migrated as long as the impact category is specified in the rateplan.

Recommended Action

For the rate plan selector which has multiple rules that result in the same rate plan with different impact categories, create a separate rate plan for each impact category. In each newly created rate plan, inspect the rate objects and remove any balance impacts that specify impact categories not applicable to that rate plan.

Example 1
Table 10–2 shows a rate plan selector with multiple rules that result in the same rate plan (rate plan A) with different impact categories:

**Table 10–2  Rate Plan Selector**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Rule Expression</th>
<th>Rate Plan</th>
<th>Impact Category Used in Rate Plan Selector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1</td>
<td>Attribute values</td>
<td>Rate plan A</td>
<td>IC1</td>
</tr>
<tr>
<td>Rule 2</td>
<td>Attribute values</td>
<td>Rate plan A</td>
<td>IC2</td>
</tr>
<tr>
<td>Rule 3</td>
<td>Attribute values</td>
<td>Rate plan A</td>
<td>*</td>
</tr>
<tr>
<td>Rule 4</td>
<td>Attribute values</td>
<td>Rate plan A</td>
<td>default</td>
</tr>
</tbody>
</table>

Table 10–3 shows the rate objects in rate plan A.

**Table 10–3  Rate Plan A**

<table>
<thead>
<tr>
<th>Rate Object</th>
<th>Impact Category Used in Rate Object</th>
<th>Balance Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Object 1</td>
<td>IC1</td>
<td>$1.00</td>
</tr>
<tr>
<td></td>
<td>IC2</td>
<td>$2.00</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>$5.00</td>
</tr>
<tr>
<td></td>
<td>default</td>
<td>$0.50</td>
</tr>
<tr>
<td>Rate Object 2</td>
<td>IC1</td>
<td>$1.50</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>$6.00</td>
</tr>
<tr>
<td>Rate Object 3</td>
<td>IC1</td>
<td>$1.75</td>
</tr>
<tr>
<td></td>
<td>default</td>
<td>$0.50</td>
</tr>
<tr>
<td>Rate Object 4</td>
<td>default</td>
<td>$0.50</td>
</tr>
</tbody>
</table>

Solution

Make the changes shown in Table 10–4 to the rate objects:

**Table 10–4  Rate Plan Selector**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Rule Expression</th>
<th>Rate Plan</th>
<th>Impact Category Used in Rate Plan Selector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1</td>
<td>Attribute values</td>
<td>Rateplan A_IC1</td>
<td>IC1</td>
</tr>
<tr>
<td>Rule 2</td>
<td>Attribute values</td>
<td>Rateplan A_IC2</td>
<td>IC2</td>
</tr>
<tr>
<td>Rule 3</td>
<td>Attribute values</td>
<td>Rateplan A_*</td>
<td>*</td>
</tr>
<tr>
<td>Rule 4</td>
<td>Attribute values</td>
<td>Rateplan A_default</td>
<td>default</td>
</tr>
</tbody>
</table>

Table 10–5 shows the rate objects in rate plan A_IC1.

**Table 10–5  Rate Plan A_IC1**

<table>
<thead>
<tr>
<th>Rate Object</th>
<th>Impact Category Used in Rate Object</th>
<th>Balance Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Object 1</td>
<td>IC1</td>
<td>$1.00</td>
</tr>
</tbody>
</table>
Table 10–6 shows the rate objects in rate plan A_IC2.

<table>
<thead>
<tr>
<th>Rate Object</th>
<th>Impact Category Used in Rate Object</th>
<th>Balance Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Object 1</td>
<td>IC2</td>
<td>$2.00</td>
</tr>
<tr>
<td></td>
<td>IC2 (previously *)</td>
<td>$5.00</td>
</tr>
<tr>
<td>Rate Object 2</td>
<td>IC2 (previously *)</td>
<td>$6.00</td>
</tr>
<tr>
<td>Rate Object 3</td>
<td>IC2 (previously default)</td>
<td>$0.50</td>
</tr>
<tr>
<td>Rate Object 4</td>
<td>IC2 (previously default)</td>
<td>$1.00</td>
</tr>
</tbody>
</table>

Table 10–7 shows the rate objects in rate plan A_IC3.

<table>
<thead>
<tr>
<th>Rate Object</th>
<th>Impact Category Used in Rate Object</th>
<th>Balance Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Object 1</td>
<td>*</td>
<td>$5.00</td>
</tr>
<tr>
<td>Rate Object 2</td>
<td>*</td>
<td>$6.00</td>
</tr>
<tr>
<td>Rate Object 3</td>
<td>* (previously default)</td>
<td>$0.50</td>
</tr>
<tr>
<td>Rate Object 4</td>
<td>* (previously default)</td>
<td>$0.50</td>
</tr>
</tbody>
</table>

Table 10–8 shows the rate objects in rate plan A_default.

<table>
<thead>
<tr>
<th>Rate Object</th>
<th>Impact Category Used in Rate Object</th>
<th>Balance Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate Object 1</td>
<td>default</td>
<td>$0.50</td>
</tr>
<tr>
<td></td>
<td>default (previously *)</td>
<td>$5.00</td>
</tr>
<tr>
<td>Rate Object 2</td>
<td>default (previously *)</td>
<td>$6.00</td>
</tr>
<tr>
<td>Rate Object 3</td>
<td>default</td>
<td>$0.50</td>
</tr>
<tr>
<td>Rate Object 4</td>
<td>default</td>
<td>$0.50</td>
</tr>
</tbody>
</table>
Derived Impact Category Used in Zone Model
Impact categories defined for zoning can be flagged as derived, which means that they can only be used in USC and APN selectors. BRM allows derived impact categories to be used in a zone model, but PDC does not. If you migrate a BRM zone model with an impact category of type derived, the migration utility reports an error.

Recommended Action
If a derived impact category is used in a zone model, change the type from 1 (derived) to 0.

Example
The impact category, Brazil, which is associated with the zone item, San Jose to Brazil, is defined as derived and is used in a zone model.

Solution
Update the IFW_IMPACT_CAT table using the following SQL statement:

```
UPDATE IFW_IMPACT_CAT SET TYPE=0 WHERE Impact_Category LIKE 'Brazil';
```

Zone Items Differentiated Only by Service Code
In PDC, a zone item in a zone model does not specify a service code. In BRM, if two zone items in a zone model differ by only the service code, then the migrated zone items gives a unique constraint error in PDC because they will be identical.

Recommended Action
Modify the zone items that differ only by service code to make them unique or remove the duplicate item.

Example
Modify the ValidFrom field in the IFW_STANDARD_ZONE table to a unique value (for example, 02-JAN-99 in the above example), to make the combination unique without using the service code.

<table>
<thead>
<tr>
<th>Origin</th>
<th>Destination</th>
<th>Valid From</th>
<th>Service</th>
<th>Zone Impact Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>456</td>
<td>1999-01-01</td>
<td>SMS1</td>
<td>IC1</td>
</tr>
<tr>
<td>123</td>
<td>456</td>
<td>1999-01-01</td>
<td>SMS2</td>
<td>IC2</td>
</tr>
</tbody>
</table>

Solution
Modify the ValidFrom field in the IFW_STANDARD_ZONE table to a unique value (for example, 02-JAN-99 in the above example), to make the combination unique without using the service code.

Extended Service Class Event Mapping
In PDC, an extended service class does not inherit the event mapping of the parent class, and all events to be used with the extended service class must be explicitly mapped. In BRM, an extended service class inherits the event mapping of the parent class. If you migrate a BRM pricing object with a service-event combination that is not available in the service-event mapping, the migration utility reports an error.

Recommended Action
Ensure that all extended service classes have mappings for all the events used in pricing objects that reference them.

Example
Fixing Errors Found in the Analysis Phase

The service-event map (**pin_event_map** file) contains these mappings:

```
/service/ip       :/event/session   ...
    :/event/session/dialup ...
/service/ip/gprs  :/event/session/gprs/master
    :/event/session/gprs/subsession ...
```

A product exists that applies to /service/ip/gprs and /event/session. The migration utility reports an error because /event/session is not mapped to /service/ip/gprs.

**Solution**

Map /service/ip/gprs to /event/session.

```
/service/ip       :/event/session   ...
    :/event/session/dialup ...
/service/ip/gprs  :/event/session/gprs/master
    :/event/session/gprs/subsession ...
    :/event/session
```

**Service Contains Deal Mapped to Parent Service**

In BRM, Pricing Center allows to add a service to a plan and to add a deal to that service which applies to a parent class of the service. PDC does not allow this configuration, and the migration utility reports an error if such a plan is migrated.

**Recommended Action**

Change the service in the plan to match the service referenced in the deal.

**Example**

Plan A applies to /service/telco/gsm/voice and contains:

- Deal 1, which applies to /service/telco.
- Deal 2, which applies to /service/telco/gsm.

**Solution**

The plan should be changed to have the following service/deal combination:

- /service/telco
  - Deal 1
- /service/telco/gsm
  - Deal 2

**Using Overlapping Date Ranges to Sequentially Impact Multiple Resources**

PDC does not support BRM rate plans with more than one date tier with overlapping date ranges. This type of configuration is typically used to first consume an included balance, such as Anytime Minutes and after the balance is exhausted, to charge for the remaining usage.

**Recommended Action**

Re-configure the product to contain only a usage charge and configure a discount to credit the charge and debit the available balance.

**Example**

A rate plan for voice calls has two date tiers:

- Consume Minutes, with a validity period that starts immediately and never ends.
- Charges for call, with a validity period that starts immediately and never ends.

**Solution**

Re-configure the product by doing the following:

1. Create a discount based on the user scenario for consumption.
2. Remove the consumption tier from the product.
3. Do one of the following:
   - Create a new deal and add the discount created in step 1.
   - Add the discount created in step 1 to the existing deal and cancel the existing deal for the subscriber.
4. Purchase the new or modified deal in BRM.

**Quantity Brackets in a Fold**

PDC requires the basis for quantity brackets in a fold to be specified as *Resource Balance* and the resource to be a non-currency resource. If you migrate a BRM fold which has basis for quantity bracket configured with anything other than *Resource Balance* or the resource is a currency resource, the migration utility reports an error.

**Recommended Action**

Correct the fold configuration to use *Resource Balance* as the basis for quantity brackets and specify a non-currency resource.

**Examples**

- The basis for quantity discount bracket is *Continuous* or *Rate Dependent*.
- The basis is *Resource Balance* but the resource is a currency resource.

**Solution**

Review the business scenario and ensure that your fold configuration is implemented correctly. You can only fold a non-currency resource and the basis for quantity brackets in a fold must be *Resource Balance*.

**BRM Object Names Containing Special Characters**

PDC does not support the character "\=" in a pricing object name. The migration utility reports an error if such objects are migrated.

**Recommended Action**

Change the pricing object name to not contain the "\=" character.

**Custom Fields**

The migration utility is not aware of any custom fields that you may have created in the BRM database.

For example, the migration utility reports the following error while converting the event storable class to XML if the event storable class has custom fields:

```
SEVERE: Error while run migration!
oracle.communications.brm.pdc.migration.MigrationException: Got a non zero exit status from the command:
/home/pin/7.5/bin/storableclasstoxml -r
/home/pin/opt/oracle/11.2.0.3/apps/migration/xml/extract/Event.xml -o /event/*. Check if the BRM is up and running.
```

Migrating Pricing Data to PDC With RRE and BRE
Recommended Action

Make custom fields available to applications. Refer to the discussion about making custom fields available to your applications for Java application in BRM Developer’s Guide.

---

**Important:** After creating the JAR file, restart the Connection Manager (CM) before running the migration utility.

---

Changes Required After Migration

The following changes are required after migration:

- Selector with Multiple RUMs
- Non-Currency Resources Used as Counters
- BRM Rate Plans with Multiple Currencies

**Selector with Multiple RUMs**

In PDC, pricing using multiple RUMs is implemented with a single charge that contains separate charge trees for each RUM. In BRM, pricing using multiple RUMs is implemented with separate rate plans or rate plan selectors for each RUM.

For BRM rate plan selectors with multiple RUMs, the migration utility does not do a complete migration of the rate plan selectors. The rate plan selector is migrated to one charge selector with one rule mapped to a single charge and the charge will contain multiple charge trees, one for each RUM.

**Recommended Action**

After migration, re-configure the charge selector in PDC by defining rules where each rule results in a charge that contains a charge tree for each RUM.

**Non-Currency Resources Used as Counters**

When configuring a credit balance impact in PDC, only positive values are accepted in the PDC UI and the value is stored as a negative value, which is the convention for credits. Therefore, the value in the PDC UI has a different sign from the corresponding value in the database.

In Pricing Center, when a non-currency balance element is impacted with a positive value, as in a counter, it is migrated as follows:

- If Grantable flag is not checked in the balance impact, it is migrated to a debit in PDC.
- If Grantable flag is checked, it is migrated to a credit with a positive value in PDC. Because PDC reverses the sign of a credit amount, the migrated credit is initially displayed as a negative value, but when **Save** is clicked, the negative sign is removed, because negative values are not accepted in the UI. Consequently, the value in the database becomes a negative value. This changes the initial configuration and would likely result in pricing errors.

**Recommended Action**

After migration, change the balance element to a counter. If the balance element is tagged as a counter, the database value is not changed and PDC displays this balance impact as an increase of the counter using the actual amount stored in the database.
Example

Given the following balance impact in Pricing Center:

- **Impact**: ‘Dollars Spent’
- **Impact**: ‘Dollars Spent’
- **Id**: 1500001
- **Amount**: 0.40 per minute
- **Grantable**: Yes

After migration, the balance impact in PDC is:

- **Credit**: ‘Dollars Spent’
- **Id**: 1500001
- **Amount**: 0.40 per minute

Without changing the balance element to a counter, if this balance impact is displayed in PDC and saved, it is stored as:

- **Credit**: ‘Dollars Spent’
- **Amount**: -0.40 per minute
- **Id**: 1500001

After changing the balance element to be a counter, if this balance impact is displayed in PDC, it is displayed and saved as:

- **Increase**: ‘Dollars Spent’
- **Id**: 1500001
- **Amount**: 0.40 per minute

Solution:

Update the BalanceElement table using the following SQL statement.

```
UPDATE BalanceElement
SET Counter=1
WHERE NumericCode=1500001;
```

**BRM Rate Plans with Multiple Currencies**

In BRM, a product can have multiple real-time rate plans, one rate plan per currency or one pipeline rate plan with multiple currencies. These product configurations are migrated in different ways.

**BRM Real-Time Rate Plans**

If you migrate a BRM product with multiple real-time rate plans per currency, it is migrated as one charge offer with one charge that has multiple branches, one for each currency charge. This is not displayed in the PDC application.

**Recommended Action**

After migration, if you want to change a price in the migrated charge offer, you can edit the XML for the charge offer, change prices for the appropriate currencies, and then import the charge offer without viewing it in a changeset in the PDC application. The changeset is stored in PDC and published to BRM.

**BRM Pipeline Rate Plans**

If you migrate a BRM product with a pipeline rate plan with multiple currencies, it is migrated as one charge offer with one charge that has multiple currencies. The currencies that are not specified for the rate plan are displayed as read-only in the PDC application.
Recommended Action
After migration, if you want, you can edit the charge and remove the currencies which are read-only and create duplicate charge offers with one charge per currency. For more information about configuring pricing for a charge, see the discussion about specifying charge details and configuring pricing in charges in PDC Creating Product Offerings.

Handling Migration Process Errors

Errors during the data migration can occur for various reasons, such as:

- Database connection problems
- Permission to write files
- The XML data does not comply with the XSD

If you encounter these types of errors, look in the migration utility’s log file, which is located in the log file directory specified in the BRM_Integration_Pack_Home/apps/migration/MigrateConfiguration.xml configuration file, for the specific details.

Resolving Java Heap Space Error

The migration utility is a Java process that runs on the Java Virtual Machine (JVM). While running the utility, you may receive “java.lang.OutOfMemoryError: Java heap space” error message on the WebLogic Managed Server. This occurs when the JVM runs out of heap space that it uses to run the utility.

The Java heap space size for the migration utility is set to 512 megabytes initial heap space and 4096 megabytes maximum heap space. In most cases, these default settings are sufficient. However, if you receive the Java heap space error, you may want to adjust the heap space sizes by adjusting the JVM parameters -Xms<size> and -Xms<size>, where -Xms<size> specifies the initial heap space and -Xmx<size> specifies the maximum heap space.

To set the Java heap space size:

1. Open the BRM_Integration_Pack_Home/apps/migration/MigrateBRMPricing shell script in a text editor.
2. Search for the following line:
   ```
   ${JAVA_HOME}/bin/java -Xms512m -Xmx4096m -XX:CompileThreshold=8000
   -XX:PermSize=128m -XX:MaxPermSize=1024m -cp ${clp} ${jvmopts}
   oracle.communications.brm.pdc.migration.Migrator $*
   ```
3. Change 512m and 4096m to appropriate heap sizes.
   For example:
   ```
   ${JAVA_HOME}/bin/java -Xms768m -Xmx4352m -XX:CompileThreshold=8000
   -XX:PermSize=128m -XX:MaxPermSize=1024m -cp ${clp} ${jvmopts}
   oracle.communications.brm.pdc.migration.Migrator $*
   ```
4. Update the Java heap space settings on the WebLogic Server. See the Oracle WebLogic Server documentation for more information.

For more information about Java heap space sizing guidelines, refer to the JVM documentation.
Restarting the Systems

After the migration is complete, restart all the BRM and PDC system components, including Pipeline Manager, WebLogic server, and the batch rating engine and real-time rating engine transformation engines.

Restarting the systems clears the data that is cached in the system memory during the migration.

Testing the Migrated Data

After migration is complete, test the migrated data before moving the data to the production systems.

For example:

- Using the PDC application, verify that you are able to create and modify pricing and configuration objects without any errors.
- Generate some usage events in BRM, then perform rating and billing of the events using the pricing data from before the migration and after.
- Compare the rating and billing results and verify that the results are the same.

Importing the Migrated Data to the Production Systems

After you have successfully completed the migration of the configuration and pricing objects on the development system, you can import the changes to your production system.

**Important:** To import changes from the development system to the production system, it is required that the BRM database on the production system must have exactly the same pricing data configurations as in the BRM database on the development system prior to the original migration.

To import the data to the production system:

1. Ensure that the PDC production system is a new installation of PDC. The PDC database cannot have any existing data.
2. Do one of the following:
   - If you are migrating the BRM pricing data to PDC, do the following:
     - Ensure that the BRM production database contains exactly the same configuration and pricing objects as in the BRM database on the development system prior to the original migration.
     - Propagate any updates that you made to the BRM data in the development system during the original migration to the BRM production system.
     - Propagate any updates that you made to the BRM data in the development system during the original migration to the BRM production system.
     - If you are migrating the BRM pricing data to PDC, ensure that the BRM production database contains exactly the same configuration and pricing objects as in the BRM database on the development system prior to the original migration.
1. If you are migrating the PDC pricing data to PDC, ensure that the PDC production database contains exactly the same configuration and pricing objects as in the PDC database on the development system prior to the original migration.

3. Propagate any updates that you made to the BRM data in the development system during the original migration to the BRM production system.

4. Setup and configure the loadpricelist and LoadIfwConfig utilities to connect to the BRM and Pipeline Manager production databases:
   a. On the development system, create a directory for the loadpricelist utility for each BRM production database.
      For example, load_price_list_hostname, where hostname is the name of the BRM production database server.
   b. Copy the contents from the loadpricelist directory installed on the production system.
   c. Ensure that the loadpricelist utility can connect to the BRM production database.
   d. Create a directory for the LoadIfwConfig utility for each Pipeline Manager production database.
      For example, LoadIfwConfig_hostname, where hostname is the name of the Pipeline Manager production database server.
   e. Copy the contents from the LoadIfwConfig directory installed on the production system (the default directory is XmlLoader).
   f. Ensure that the LoadIfwConfig utility can connect to the Pipeline Manager production database.

5. Copy the client.jks from the target PDC production system into the source PDC system.

6. Update the BRM_Integration_Pack_Home/apps/bin/MigrateBRMPricing file with the keystore information for the target PDC production system where you want to import the changes:
   a. On the PDC development system, copy the BRM_Integration_Pack_Home/apps/bin/MigrateBRMPricing file and rename it Hostname_MigrateBRMPricing, where Hostname identifies the target PDC system.
   b. Edit the Hostname_MigrateBRMPricing file to reference the target production systems:
      a. Search for the jvmsystems entry.
      b. Set the -Djavax.net.ssl.trustStore entry to the location of the client.jks file copied from Target server.
      c. Save and close the file.

7. Copy the BRM Integration Pack wallet from the target PDC production system into the source PDC system.

8. Update the migration utility’s configuration file with the connection information for the PDC and BRM production systems where you want to import the changes:
   a. On the development system, copy the BRM_Integration_Pack_Home/apps/migration/MigrationConfiguration.xml file and rename it
Hostname_MigrateConfiguration.xml, where Hostname identifies the target
PDC system.

b. Edit Hostname_MigrateConfiguration.xml to reference the target production
systems.

a. Search for walletConfiguration.

b. Set the pdcWalletLocation entry to the location of the BRM Integration
Pack wallet copied from the target PDC production system.

c. Search for pricingServer:

   <pricingServer>
     <connectionInfo>
       <hostName>PricingServerHostName</hostName>
       <port>PricingServerPort</port>
       <adminUser>AdminUserName</adminUser>
       <pdcUser>PDCUser</pdcUser>
     </connectionInfo>
   </pricingServer>

d. Edit the connectionInfo entry within the pricingServer element to
reference the PDC production system.

e. Search for the xrefDatabase element.

   <xrefDatabase>
     <connectionInfo>
       <login>CrossRefUserName</login>
       <hostName>CrossRefHostName</hostName>
       <port>CrossRefPort</port>
       <serviceName>CrossRefServiceName</serviceName>
     </connectionInfo>
   </xrefDatabase>

f. Edit the connectionInfo within the xrefDatabase element to reference the
PDC transformation database.

g. Search for the targetBRM element.

h. Edit the loadPriceListDir within the targetBRM element to reference the
directory where loadpricelist utility is run to update the BRM system.

i. (Only if Pipeline data was migrated by the original migration) Edit the
loadIfwConfigDir within the targetBRM element to reference the
directory where LoadIfwConfig utility is run to update the Pipeline
database.

9. Run the following command, which starts the import of the changes to PDC and
BRM production systems:

   MigrateBRMPricing -retarget -properties Hostname_MigrateConfiguration.xml

   If the retarget migration process stops, run the following command, which restarts
the process:

   MigrateBRMPricing -retarget -restart -properties Hostname_
   MigrateConfiguration.xml

When the retarget migration process is complete, a status report is generated in the
report directory location specified in the BRM_Integration_Pack_
Home/apps/migration/MigrateConfiguration.xml configuration file. The status
report file name includes the target PDC system host name and port number. For example, `StatusReport_Retarget_hostname_portnumber.html`.

10. Remove the XML files generated during migration (Optionally, you can store them in a secure location if you need them for future use).

   a. On the development system, open the `BRM_Integration_Pack_Home/apps/migration/MigrateConfiguration.xml` file.

   b. Search for the `brmExtractedXML` element:

   ```xml
   <brmExtractedXML>BRMExtractedXML</brmExtractedXML>
   ```

   c. Remove the XML files in the directory location specified by `BRMExtractedXML`.

   d. Search for the `brmXML` element:

   ```xml
   <brmXML>BRMDataFile</brmXML>
   ```

   e. Remove the XML files in the directory location specified by `BRMDataFile`.

   f. Search for the `pdcXML` element:

   ```xml
   <pdcXML>PDCDataFile</pdcXML>
   ```

   g. Remove the XML files in the directory location specified by `PDCDataFile`.

---

**About Working with the Pricing Data after Migration**

After the BRM pricing data has been successfully migrated to PDC, you use the PDC application to modify the pricing data or create new pricing configurations. PDC transforms the data into the BRM pricing data model and then loads the data into the BRM database.

Modifications to the configuration objects which are mastered in BRM must be done in BRM and then synchronized with PDC by using `SyncPDC` utility. See the discussion about synchronizing setup components in *PDC Creating Product Offerings* for more information.
This chapter describes the Oracle Communications Pricing Design Center (PDC) Migration Reports.

For information about pricing data migration, see "Migrating Pricing Data to PDC With ECE" and "Migrating Pricing Data to PDC With RRE and BRE".

**About Migration Reports**

For every migration process, PDC migration utility generates the following reports at different stages:

- **Analysis Reports**
- **Compatibility Reports**
- **Migration Report**
- **Process Report**
- **Reconciliation Reports**
- **Status Report**

The migration reports are stored in the report directory location specified in the `BRM_Integration_Pack_Home/apps/migration/MigrateConfiguration.xml` configuration file.

**Analysis Reports**

The analysis reports are generated during the data analysis phase of the migration. The reports list the data configurations that are not supported in PDC and that will be modified by the migration utility during the migration process and any data configurations that you need to change manually. Each object in the report is associated with an error message that describes the reason for the error and how to fix it and one of the following severity types:

- **Critical**: Indicates that the object configuration is invalid or not supported in PDC. You must manually fix these object configurations in the BRM database.

- **User input**: Indicates that the migration utility requires more information about the object configuration to make adjustments. You must provide the required information in XML format.

- **Warning**: Indicates that the object will be skipped and not migrated to PDC or the data will be migrated to PDC with changes to some attributes.

- **Info**: Indicates that the object will be modified by the migration utility during the migration process.
For the configuration migration process, only one analysis report is generated.

For the pricing migration process, separate analysis reports are generated for the product, discount, sponsorship, and subscription objects. You need to review all four analysis reports to fix any critical or user input errors. The reports include links to detailed reports that contain the list of the BRM objects with errors.

Figure 11–1 shows a sample product analysis output for the rate plan object.

**Figure 11–1 Sample Product Analysis Output for Rate Plan Object**

**Analysis Report for MIGRATE_RATEPLAN, Batch 1. Generated on 20131204_072317_405**

**Warning** - Migration will skip the entities marked as warning  
**Critical** - Migration will not proceed without the errors being resolved  
**User Input** - User has to provide input to resolve the conflicts

<table>
<thead>
<tr>
<th>BRM Code</th>
<th>BRM Name</th>
<th>Analysis Message</th>
<th>Severity Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>RADJSDLMS</td>
<td>Rate adjust and special day link for multiple services</td>
<td>The charge rate plan, &quot;RADJSDLMS&quot;, has configurations for the service codes, &quot;SMS&quot;, which are not used in any product. These configurations will not be migrated.</td>
<td>Warning</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>The rate plan has versions which have Rate Adjusts and/or Special Day Links defined. PDC does not support these configurations. You can use Discounting and Holiday Calendar to achieve similar functionality. Delete the Rate Adjust and or Special Day Link configurations in the rate plan versions from BRM and restart the pricing job.</td>
<td>Critical</td>
</tr>
</tbody>
</table>

**BRM BRE RatePlans Analyzed: 1**

**BRM BRE RatePlans Can Be Migrated: 0**

**BRM BRE RatePlans Will Be Skipped: 0**

**BRM BRE RatePlans With Critical Errors: 1**

**Compatibility Reports**

The compatibility reports are generated during the data migration phase of the migration. The reports list the data configurations that are not supported in ECE and that will be modified by the migration utility during the migration process and any data configurations that you need to change manually. Each object in the report is associated with an error message that describes the reason for the error and how to fix it and one of the following severity types:

- **Critical**: Indicates that the object configuration is invalid or not supported in ECE. You must manually fix these object configurations in the BRM or PDC database.
- **User input**: Indicates that the migration utility requires more information about the object configuration to make adjustments. You must provide the required information in XML format.
- **Warning**: Indicates that the object will be skipped and not migrated to PDC or the data will be migrated to PDC with changes to some attributes.
- **Info**: Indicates that the object will be modified by the migration utility during the migration process.

For the configuration migration process, only one compatibility report is generated.

For the pricing migration process, separate compatibility reports are generated for the product, discount, sponsorship, and subscription objects. You need to review all four compatibility reports to fix any critical or user input errors. The reports include links to detailed reports that contain the list of the BRM or PDC objects with errors.

### Migration Report

A migration report is generated at the end of a migration process.

For each object type in the migration process, the report lists the BRM object name, PDC object name, PDC internal ID, transformation message, and severity.

In addition, the report provides the following information:

- The total number of BRM or PDC objects
- The total number of objects migrated
- The total number of objects skipped

Only one migration report is generated for the configuration migration process and separate reports for the product, discount, sponsorship, and subscription objects are generated for the pricing migration process.

### Process Report

A process report is generated after the migration utility completes the BRM pricing object modifications for transformation to the PDC data model.

The report lists all the pricing objects that were modified, created, or duplicated by the migration utility.

---

**Note:** A process report is not generated for the subscription objects.

### Reconciliation Reports

The reconciliation reports are generated at the end of the pricing migration process.

Separate reconciliation reports for product, discount, sponsorship, and subscription objects are generated.

The report provides the following information:

- The number of BRM or PDC objects before migration
- The number of BRM objects created in the source database
- The number of BRM objects modified in the source database
- The total number of BRM objects migrated

### Status Report

A status report is generated after you run the migration utility to import data to the production systems.
This chapter provides reference information for the Oracle Communications Billing and Revenue Management (BRM) pricing migration utility.
MigrateBRMPricing

Use the MigrateBRMPricing utility to migrate existing pricing and configuration objects from the BRM database to the PDC database.

MigrateBRMPricing uses the PDC_home/apps/migration/MigrationConfiguration.xml file (where PDC_home is the directory in which PDC is installed) that contains login information for the databases and other configuration settings.

See "Migrating Pricing Data to PDC With ECE" and "Migrating Pricing Data to PDC With RRE and BRE" for more information.

Location

PDC_home/apps/bin

Syntax

Start migration process:

MigrateBRMPricing -object -analyze

Resume migration process:

MigrateBRMPricing -resume

Restart migration process:

MigrateBRMPricing -object -analyze -restart

Retarget migration:

MigrateBRMPricing -retarget -properties MigraeConfiguration_FileName

Generate an analysis report:

MigrateBRMPricing -object -analyze -report

Generate a status report:

MigrateBRMPricing -status

Skip critical errors encountered during migration:

MigrateBRMPricing -pricingObject -skip

Parameters

-object

Specifies the BRM objects to migrate. The object must be one of the following:

- config
- pricing
- product
- discount
- sponsorship
• subscription

-pricingObject
Specifies the BRM pricing objects to migrate. The pricingObject must be one of the following:
• pricing
• product
• discount
• sponsorship

-analyze
Extracts BRM objects from the BRM database and performs analysis of the data.
For example, the following command extracts and analyzes the configuration objects:
MigrateBRMPricing -config -analyze

-resume
Resumes the migration process that is currently in progress.

-restart
Restarts the migration process.
For example, the following command restarts the configuration objects migration:
MigrateBRMPricing -config -analyze -restart

-retarget
Retargets the migration data to another PDC and BRM system.

---

Note: Before running the migration utility with the -retarget parameter, update the MigrateConfiguration.xml file to the PDC and BRM systems where you want to retarget the migration data.

---

-properties MigrateConfiguration_FileName
Specifies the name and location of the migration configuration file that is used with the -retarget parameter.

-report
Generates an analysis report.
For example, the following command generates the configuration analysis report:
MigrateBRMPricing -config -analyze -report

-status
Generates a status report.

---

Note: You cannot use the -status parameter with -analyze, -report, or -restart parameter.

---

-skip
Skips the critical errors encountered for the pricing objects and continues the migration process.
For example, the following command skips the discount objects if there are critical errors encountered for these objects during the migration analysis phase and continues the migration:

```
MigrateBRMPricing -discount -skip
```

### Results

If the utility displays an error message, look in the `MigrateBRMPricing` log file to determine the cause of the error. The log file is located in the log file directory specified in the migration configuration file.