

Oracle® Communications Billing and Revenue Management

Upgrade Guide



Release 12.0

E51037-09

January 2022

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

E51037-09

Copyright © 2017, 2022, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	x
Documentation Accessibility	x
Diversity and Inclusion	x

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	1-3
Identifying Who Is Affected by the Upgrade	1-3
Collecting Information about Your System	1-3
Determining the Impact of New Features	1-4
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?	1-5
Updating Your System Environment	1-5
Creating Test Environments	1-5
Old Baseline Release	1-6
Old Customized Release	1-6
New Baseline Release	1-6
New Customized Release (Test System)	1-6
Transferring Customizations to the New Release	1-7
Upgrading Customized Policy Source Files	1-7
Updating Configuration Files	1-8
Updating Database Customizations	1-8
Modifying the Content of Custom Tables	1-8
Modifying the Structure of Custom Tables	1-8
Fixing Standard Database Objects with Nonstandard Object IDs	1-9
Updating Custom Reports	1-9

Updating Custom Applications	1-9
Testing Your Upgraded System	1-9
Running Old and New Versions in Parallel	1-10
Testing the BRM Database	1-10
Troubleshooting Your Upgraded System	1-10
Testing Checklist	1-11
Preparing for the Production System Upgrade	1-12
Upgrading Your Production System	1-12

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	2-2
Pricing Center	2-2
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	2-3
AAA Managers Not Supported in BRM 12.0	2-3
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
Utility Log File Permissions Changed	2-3
French Direct Debit Not Supported in 12.0	2-4
Changed Registry.js Keys in Billing Care SDK	2-4
Billing Care SDK Requires CSRF Tokens in AJAX Requests	2-4
PCM SDK is Now Part of BRM SDK	2-4
SNMP Utilities Not Supported in 12.0	2-4
Storing Passwords in Oracle Wallet	2-5

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
Changed 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	8-2
Preparing for the Upgrade to BRM 12.0 or 12.0 Patch Sets	8-3
Shutting Down the Current Instance	8-3
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	8-4
Backing Up Your BRM 7.5 Database	8-4
Creating Your BRM 12.0 Database	8-5
Upgrading to BRM 12.0 or 12.0 Patch Sets	8-5
Importing Your BRM 7.5 Data	8-5
Installing Java	8-6
Installing Perl on Linux	8-6

Installing Perl on Solaris	8-7
Obtaining the ojdbc JAR File	8-8
Setting the Environment Variables	8-8
Setting BRM Wallet Location in sqlnet.ora	8-9
Downloading the BRM Software	8-9
Installing BRM and Pipeline Manager for Upgrade	8-9
Configuring BRM 12.0 Patch Set 3 or Later to Use the AES Encryption Method	8-12
Upgrading Your Data to BRM 12.0	8-14
Pointing the BRM 12.0 Installation to the Database Schema	8-14
Upgrading the BRM Database Schema to the BRM 12.0 Schema	8-14
Upgrading the schema on single-schema systems	8-14
Upgrading the Schema on Multischema Systems	8-16
Upgrading the Pipeline Manager Database Schema to the BRM 12.0 Schema	8-19
Completing the Upgrade to BRM 12.0 or 12.0 Patch Sets	8-20
Installing the BRM 12.0 Client Applications	8-20
Adding Optional Components	8-21
Updating Configurations	8-21
Update DM_IFW_SYNC Queue Database Configuration	8-21
Verify the Account Synchronization Queue in the Database	8-21
Update the Default Settings for the pin_rel Utility	8-22
Updating the Settings for the pin_virtual_time Utility	8-22
Upgrading AMM in a Multischema System	8-22
Restoring Customizations	8-23
Post-Upgrade Procedures	8-23
Creating an Oracle AQ Database Queue	8-23
Loading the pin_notify File	8-24
Copying Files for Compiling CMs	8-24
Setting the Environment Variables for Pipeline Manager	8-25
Configuring SSL for the BRM Database	8-25
Running the create_procedures Script	8-25
Merging the Price Lists	8-26
Setting the DM_IFW_SYNC Queue Database and CM Configuration Entries	8-26
Restoring Service to Your Customers	8-27

9 Upgrading Business Operations Center

About Upgrading Business Operations Center	9-1
Downloading Business Operations Center	9-1
Upgrading Business Operations Center to the Latest Patch Set Release	9-2
Upgrading Your Business Operations Center Database Schema Manually	9-4

10 Migrating Pricing Data to PDC With ECE

About Migrating BRM Pricing Data to PDC With ECE	10-2
About the BRM Data Analysis Phase for Migrating BRM Pricing Data to PDC With ECE	10-3
About the Data Migration Phase for Migrating BRM Pricing Data to PDC With ECE	10-4
Performing Migration of BRM Pricing Data to PDC With ECE	10-5
Creating a BackUp of Your BRM Database	10-6
Setting Up the BRM Development System	10-6
Configuring the Migration Utility for PDC with ECE	10-8
Migrating the BRM Pricing Data to PDC With ECE	10-14
Migrating BRM Configuration Objects	10-15
Migrating BRM Pricing Objects	10-15
About Migrating PDC Pricing Data to PDC With ECE	10-22
About the Data Migration Phase for Migrating PDC Pricing Data to PDC With ECE	10-22
Performing Migration of PDC Pricing Data to PDC With ECE	10-23
Creating a BackUp of Your PDC Database	10-24
Setting Up the PDC Development System	10-24
Migrating the PDC Pricing Data to PDC With ECE	10-25
Migrating PDC Configuration Objects	10-25
Migrating PDC Pricing Objects	10-26
Fixing Errors Found in the Analysis Phase	10-30
Fixing Analysis "User Input Required" Errors	10-30
Identifying Resource ID for Noncurrency Resources Used in Batch Rating	10-30
Providing Values List for Expressions in Discount and Chargeshare Configurations	10-30
Analysis Errors Requiring Pricing Data Changes	10-31
Multiple Impact Categories in a Rate Plan	10-31
Derived Impact Category Used in Zone Model	10-34
Zone Items Differentiated Only by Service Code	10-34
Extended Service Class Event Mapping	10-35
Service Contains Deal Mapped to Parent Service	10-35
Using Overlapping Date Ranges to Sequentially Impact Multiple Resources	10-36
Quantity Brackets in a Fold	10-36
BRM Object Names Containing Special Characters	10-37
Custom Fields	10-37
Fixing Errors Found in the Migration Phase	10-37
Skipping Errors and Continue Pricing Objects Migration	10-38
Fixing Compatibility "User Input Required" Errors	10-38
Identifying Taxcode for the General Ledger ID Used in Offline Charging	10-39

Mapping Batch Rating Engine Fields Used in Selectors to Corresponding ECE Fields	10-39
Mapping Custom Rules To Usage Types in Selectors	10-40
Identifying ECE Fields for RUM Fields in BRM	10-41
Compatibility Errors Requiring Pricing Data Changes	10-41
Events Split across Time Periods	10-42
Quantity Ranges With Negative Values	10-42
Changes Required After Migration	10-42
Rate Plan Selector with Multiple RUMs	10-42
Rate Plan Selector with Zone Models	10-43
Noncurrency Resources Used as Counters	10-43
BRM Rate Plans with Multiple Currencies	10-44
Handling Migration Process Errors	10-44
Resolving Java Heap Space Error	10-45
Restarting the Systems	10-45
Testing the Migrated Data	10-45
Importing the Migrated Data to the Production Systems	10-46
About Working with the PDC Pricing Data after Migration	10-48

11 Migrating Pricing Data to PDC With RRE and BRE

About Migrating BRM Pricing Data to PDC With RRE and BRE	11-1
About the Data Analysis Phase for Migrating BRM Pricing Data to PDC With RRE and BRE	11-3
About the Data Migration Phase for Migrating BRM Pricing Data to PDC With RRE and BRE	11-4
Performing Migration of BRM Pricing Data to PDC With RRE and BRE	11-5
Creating a BackUp of Your BRM Database	11-6
Setting Up the BRM Development System	11-6
Configuring the Migration Utility for PDC with RRE and BRE	11-8
Migrating the BRM Pricing Data to PDC With RRE and BRE	11-14
Migrating Configuration Objects	11-14
Migrating Pricing Objects	11-15
Migrating All Pricing Objects	11-15
Migrating Pricing Objects by Category	11-16
Fixing Errors Found in the Analysis Phase	11-18
Fixing Analysis "User Input Required" Errors	11-19
Identifying Resource ID for Noncurrency Resources Used in Batch Rating	11-19
Providing Values List for Expressions in Discount and Chargeshare Configurations	11-19
Analysis Errors Requiring Pricing Data Changes	11-20
Multiple Impact Categories in a Rate Plan	11-20
Derived Impact Category Used in Zone Model	11-23

Zone Items Differentiated Only by Service Code	11-23
Extended Service Class Event Mapping	11-24
Service Contains Deal Mapped to Parent Service	11-24
Using Overlapping Date Ranges to Sequentially Impact Multiple Resources	11-25
Quantity Brackets in a Fold	11-25
BRM Object Names Containing Special Characters	11-26
Custom Fields	11-26
Changes Required After Migration	11-26
Selector with Multiple RUMs	11-26
Noncurrency Resources Used as Counters	11-27
BRM Rate Plans with Multiple Currencies	11-28
Handling Migration Process Errors	11-28
Resolving Java Heap Space Error	11-29
Restarting the Systems	11-29
Testing the Migrated Data	11-29
Importing the Migrated Data to the Production Systems	11-30
About Working with the Pricing Data after Migration	11-33

12 Migration Reports

About Migration Reports	12-1
Analysis Reports	12-1
Compatibility Reports	12-2
Migration Report	12-3
Process Report	12-3
Reconciliation Reports	12-3
Status Report	12-4

13 BRM Pricing Migration Utility

MigrateBRMPricing	13-1
-------------------	------

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.

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

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

Part I

About Upgrading

This part contains a general overview about upgrading your Oracle Communications Billing and Revenue Management (BRM) software.

This part contains the following chapters:

- [About Upgrading BRM Releases](#)

1

About Upgrading BRM Releases

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

Topics in this document:

- [About Upgrading BRM to a New Release](#)
- [Planning Your Upgrade](#)
- [Updating Your System Environment](#)
- [Creating Test Environments](#)
- [Transferring Customizations to the New Release](#)
- [Testing Your Upgraded System](#)
- [Preparing for the Production System Upgrade](#)
- [Upgrading Your Production System](#)

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.



Note:

- 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 about the BRM 7.5 releases from which you can directly upgrade to BRM 12.0 or 12.0 patch sets, see "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 *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 your 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.
- 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
- 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:

- For a list of hardware and software supported by BRM, see "Hardware and Software Requirements" in *BRM Installation Guide*.
- For general database information, see "Database Configuration and Tuning" in *BRM Installation Guide*.

- Check disk space and memory.

Ensure that your test environments and production system include the disk space and memory required for the new BRM release. The requirements might differ from the requirements for the old release. For more information, see the following:

- For disk space requirements, see "Hardware and Software Requirements" in *BRM Installation Guide*.
- For information on memory configuration, see "Improving BRM Performance" in *BRM System Administrator's Guide*.

- Tune your system.

Upgrading is faster and easier if you *first* tune your system for optimal performance. For assistance with estimating the hardware and storage requirements for your BRM system, contact Oracle Support for more information.

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](#)".

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.

 **Note:**

To upgrade the old data to the new format, you should create custom SQL scripts and incorporate them into the upgrade configuration file, **upgrade.cfg**, *before running the database upgrade scripts*.

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.



Note:

To identify undocumented database customizations in your old release, compare the database schema in the "[Old Baseline Release](#)" with the database schema in your "[Old Customized Release](#)".

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.



Note:

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.



Note:

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.

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.



Note:

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.
- **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.

Note:

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.

Caution:

Do not begin your upgrade until you have backed up 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 \(Test System\)](#)".

Part II

Upgrade Impacts From 7.5 to 12.0

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](#)

2

Feature Changes from BRM 7.5 to BRM 12.0

Learn about the feature changes that affect your Oracle Communications Billing and Revenue Management (BRM) system and what you must consider when you upgrade from 7.5 to 12.0.

Topics in this document:

- [BRM Clients](#)
- [Changes to Supported Software Versions](#)
- [Content Manager](#)
- [Optional Managers](#)
- [Service Managers](#)
- [Voucher Manager](#)
- [Web Services Manager](#)
- [System Administration](#)

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 Compatibility Matrix*.

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.

Utility Log File Permissions Changed

To increase security, the permissions for utility log files has been changed to **0640**.

- BRM 7.5 utility log file permissions: **0664**

- BRM 12.0 utility log file permissions: **0640**

French Direct Debit Not Supported in 12.0

BRM 12.0 no longer supports French Direct Debit payment processing.

Changed Registry.js Keys in Billing Care SDK

In the Billing Care SDK **registry.js** file, all instances of the **viewModel** key were changed to **viewmodel**.

If your BRM 7.5 system includes custom **registry.js** files, update all instances of the **viewModel** key to **viewmodel**. For more information, see the latest **registry.js** file packaged with the BRM 12.0 release.

Billing Care SDK Requires CSRF Tokens in AJAX Requests

To improve security, the Billing Care SDK now requires all AJAX requests to include cross-site request forgery (CSRF) tokens.

If you are upgrading from BRM 7.5 to BRM 12.0 Patch Set 3 or later, you must add CSRF tokens to all of your custom AJAX requests sent to the Billing Care SDK. Without the token, the Billing Care SDK will not authorize your custom AJAX requests.

The following shows sample code for adding a CSRF token to an AJAX request:

```
$.ajax({
  type: "GET",
  dataType: "json",
  url: urlToFetch,
  beforeSend: function (xhr) {
    util.setRequestHeader(xhr)
  },
  contentType: "application/json; charset=utf-8"
}).done(function (data) {

}).fail(function (errorThrown) {

}).always(function () {

});
```

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*.

3

Storable Class Changes from BRM 7.5 to BRM 12.0

Learn about the storable class changes that affect your upgrade from Oracle Communications Billing and Revenue Management (BRM) 7.5 to BRM 12.0.

Topics in this document:

- [New Storable Classes](#)
- [Changed Storable Classes](#)

For information about planning your upgrade implementation, such as setting up your development and test environments, see "[About Upgrading BRM Releases](#)".

New Storable Classes

No new storable classes were added in BRM 12.0.

Changed Storable Classes

No storable classes were changed in BRM 12.0.

4

Opcode Changes from BRM 7.5 to BRM 12.0

Learn about the opcode changes that affect your upgrade from Oracle Communications Billing and Revenue Management (BRM) 7.5 to BRM 12.0.

Topics in this document:

- [New Opcodes](#)
- [Changed Opcodes](#)
- [Obsoleted Policy Opcodes](#)
- [Obsoleted Standard Opcodes](#)

For information about planning your upgrade implementation, such as setting up your development and test environments, see "[About Upgrading BRM Releases](#)".

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_OP_CONTENT_POL_ACCOUNTING
- PCM_OP_CONTENT_POL_AUTHORIZE
- PCM_OP_CONTENT_POL_POST_ACCOUNTING
- PCM_OP_CONTENT_POL_POST_AUTHORIZE
- PCM_OP_GPRS_AAA_POL_ACC_ON_OFF_SEARCH
- PCM_OP_GPRS_AAA_POL_AUTHORIZE
- PCM_OP_GPRS_AAA_POL_AUTHORIZE_PREP_INPUT
- PCM_OP_GPRS_AAA_POL_REAUTHORIZE_PREP_INPUT
- PCM_OP_GPRS_AAA_POL_SEARCH_SESSION
- PCM_OP_GPRS_AAA_POL_STOP_ACCOUNTING_PREP_INPUT
- PCM_OP_GPRS_AAA_POL_UPDATE_ACCOUNTING_PREP_INPUT
- PCM_OP_GSM_AAA_POL_ACC_ON_OFF_SEARCH
- PCM_OP_GSM_AAA_POL_AUTHORIZE
- PCM_OP_GSM_AAA_POL_AUTHORIZE_PREP_INPUT

- PCM_OP_GSM_AAA_POL_POST_PROCESS
- 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_AUTHORIZER
- 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_AUTHORIZER
- 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_AUTHORIZER
- 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_DETECT_CONTINUATION_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_UPDATE_AND_REAUTHORIZE
- 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

5

Utility Changes from BRM 7.5 to BRM 12.0

Learn about the utility changes that affect your upgrade from Oracle Communications Billing and Revenue Management (BRM) 7.5 to BRM 12.0.

Topics in this document:

- [New Utilities](#)
- [Changed Utilities](#)

For information about planning your upgrade implementation, such as setting up your development and test environments, see "[About Upgrading BRM Releases](#)".



Note:

To increase security, the permissions for all utility log files has been changed to **0640**.

- BRM 7.5 utility log file permissions: **0664**
- BRM 12.0 utility log file permissions: **0640**

New Utilities

[Table 5-1](#) lists the utilities that were added in BRM 12.0.

Table 5-1 New Utilities in BRM 12.0

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

Changed Utilities

[Table 5-2](#) contains a list of the utilities that were changed in BRM 12.0.

Table 5-2 Changed Utilities

Changed Utility	Description
<code>pin_crypt_app</code>	Modified to support only Oracle ZT PKI encryption algorithm for encryption and to store or retrieve configuration entries in the Oracle wallet.

6

Notification Event Changes from BRM 7.5 to BRM 12.0

Learn about the notification event changes that affect your upgrade from Oracle Communications Billing and Revenue Management (BRM) 7.5 to BRM 12.0.

Topics in this document:

- [New Notification Events](#)
- [Changed Notification Events](#)

For information about 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.

7

Module and iScript Changes from BRM 7.5 to 12.0

Learn about the module and iScript 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.

Topics in this document:

- [New Modules](#)
- [Changed Modules](#)
- [New iScripts](#)
- [Changed iScripts](#)

For information about 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.

Part III

Upgrade Procedures

This part contains information on the upgrade procedures for different categories of Oracle Communications Billing and Revenue Management (BRM) software.

This part contains the following chapters:

- [Upgrading BRM and Pipeline Manager](#)
- [Upgrading Business Operations Center](#)

8

Upgrading BRM and Pipeline Manager

Learn about the procedures required to upgrade from Oracle Communications Billing and Revenue Management (BRM) 7.5 to BRM 12.0 or to a BRM 12.0 patch set.

Topics in this document:

- [About the Upgrade](#)
- [Tasks Involved in the BRM 7.5-to-12.0 Upgrade Process](#)
- [Preparing for the Upgrade to BRM 12.0 or 12.0 Patch Sets](#)
- [Creating a Complete Backup of Your BRM 7.5 Data](#)
- [Creating Your BRM 12.0 Database](#)
- [Upgrading to BRM 12.0 or 12.0 Patch Sets](#)
- [Upgrading Your Data to BRM 12.0](#)
- [Completing the Upgrade to BRM 12.0 or 12.0 Patch Sets](#)
- [Post-Upgrade Procedures](#)
- [Restoring Service to Your Customers](#)

Note:

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 about the BRM 7.5 releases from which you can directly upgrade to BRM 12.0 or to a BRM 12.0 patch set, see "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 from BRM 7.5 to BRM 12.0 or to a BRM 12.0 patch set.

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 tasks required to upgrade from BRM 7.5 to BRM 12.0 or to a BRM 12.0 patch set. 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.

Complete these tasks 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. Create 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 about installing Perl on Linux, see "[Installing Perl on Linux](#)".
 - For instruction about installing Perl on Solaris, see "[Installing Perl on Solaris](#)".
8. Obtain the **ojdbc8.jar** jar file. See "[Obtaining the 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. Obtain the BRM software. See "[Downloading the BRM Software](#)".
12. Install the upgrade package. See "[Installing BRM and Pipeline Manager for Upgrade](#)".
13. (Optional) Configure your system to continue using AES encryption. See "[Configuring BRM 12.0 Patch Set 3 or Later to Use the AES Encryption Method](#)".
14. Point your BRM 12.0 installation to the new database schema. See "[Pointing the BRM 12.0 Installation to the Database Schema](#)".
15. Upgrade the BRM (Portalbase) database schema to the BRM 12.0 schema. See "[Upgrading the BRM Database Schema to the BRM 12.0 Schema](#)".
16. 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](#)".

17. Install BRM 12.0 client applications. See "[Installing the BRM 12.0 Client Applications](#)".
18. Add optional components. See "[Adding Optional Components](#)".
19. Check other database configurations. See "[Updating Configurations](#)".
20. Upgrade Account Migration Manager for multi-schema systems. See "[Upgrading AMM in a Multischema System](#)".
21. 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



Note:

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

To shut down BRM 7.5:

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



Note:

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.

 **Note:**

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

 **Note:**

Back up all 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

 **Note:**

- 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 about the **exp** utility, see the discussion on export and import utilities in the appropriate version of the *Oracle Database Utilities* document.

 **Note:**

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

 **Note:**

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

 **Note:**

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

After you have successfully created a new schema, create a BRM database user (which is the 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](#)". When you import the backup data, 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 about the **imp** utility, see the discussion on export and import utilities in the appropriate version of the *Oracle Database Utilities* document.

After you complete the import process, make sure that there are no invalid objects in the database schema. If there are any invalid objects, try to compile them.

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 set software pack.

See "Additional BRM Software Requirements" in *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 commands:

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

Note:

If you use a proxy server as an intermediary for Internet access, configure a proxy server to access the yum repository. See https://docs.oracle.com/cd/E37670_01/E37355/html/ol_yum_config.html#ol_proxy_config.

2. Set the environment variable for installing Perl by running the following commands:
 - 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 "Additional BRM Software Requirements" in *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 commands:

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

where *version* is the compatible version of Perl.

3. Run the following commands:

```
cd perl-version
sh Configure -des -Dcc="gcc -m32" -Dusetthreads -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:

```
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 Software Compatibility" in *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:

```
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 commands:

- 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 Solaris:

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

See "Additional BRM Software Requirements" in *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 commands:

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

where *version* is the compatible version of Perl.

3. Run the following commands:

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

where:

- *Oracle_dev_path* is the path to the directory in which 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 the ojdbc JAR File

Obtain the **ojdbc8.jar** file and save it in a directory on the machine in which BRM 12.0 is going to be installed. Note the path to this directory; you are required to specify this path in the Specify Prerequisite Libraries Location screen during BRM installation.

You can download this file from the Oracle Database 12.2.0.1 JDBC Driver & UCP Downloads page on the Oracle website.

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_241`.

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_30_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.

Downloading the BRM Software

You download the BRM software from the following locations:

- For the 12.0 release: The Oracle software delivery website (<https://edelivery.oracle.com>).
- For a 12.0 Patch Set release: The Oracle support website (<https://support.oracle.com>).

Search for and download the **Oracle Communications Billing and Revenue Management 12.0.0.x.0** software, where *x* is **0** for the 12.0 release, **1** for Patch Set 1, **2** for Patch Set 2, and so on. The package includes the following BRM installer file, where *platform* is **linux** or **solaris**.

- For the 12.0 release: `brmserver_12.0.0.0.0_platform_generic.jar`
- For a 12.0 Patch Set release: `brmserver_12.0.0.x.0_platform_generic_full.jar`

Installing BRM and Pipeline Manager for Upgrade

To install BRM and Pipeline Manager for upgrade:

1. Download the BRM software. See "[Downloading the BRM Software](#)".
2. Go to the directory where you downloaded the BRM installer, 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 the BRM 12.0 release: **brmserver_12.0.0.0.0_platform_generic.jar**

For a BRM 12.0 Patch Set:

brmserver_12.0.0.x.0_platform_generic_full.jar

where *x* is the patch set version number, and *platform* is **linux** or **solaris**.

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

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

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

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

```
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:

```
Java_home/bin/java -jar jar_file -record -destinationFile path
```

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

The Welcome screen appears.

3. Click **Next**.

The Installation Location screen appears.

4. Enter the full path or browse to the directory in which BRM 12.0 is going to be installed (if you are upgrading to BRM 12.0) or in which BRM 12.0 is already installed (if you are upgrading to a 12.0 Patch Set). Click **Next**.

The Installation Type screen appears.

5. Select **Custom**, and then click **Next**.

The Feature Sets Selection screen appears.

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

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

 **Note:**

- Ensure that you select only the components in **Upgrade Manager Framework** and the components that you already installed in BRM 7.5 or in a BRM 7.5 patch set. 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.

If you selected only the components in **Upgrade Manager Framework**, proceed to step [10](#).

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

7. Specify the information in [Table 8-1](#), and then click **Next**.

Table 8-1 Specify Prerequisite Libraries Location

Field	Description
Prerequisite Libraries	Enter the full path or browse to the directory in which the prerequisite libraries are stored.
Enable SSL for BRM server	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 for BRM server check box selected.

The Oracle ZT PKI Encryption screen appears.

8. Do one of the following, and then click **Next**.
 - If you have enabled Oracle ZT PKI Encryption for BRM, select **Yes**.
 - If you have not enabled Oracle ZT PKI Encryption for BRM, select **No** and proceed to step [10](#).

The Specify BRM Wallet Details screen appears.

9. Enter the information in [Table 8-2](#), and then click **Next**.

Table 8-2 Specify BRM Wallet Details

Field	Description
Wallet Password	Enter the password for the BRM wallet.
Wallet Location	Enter the full path or browse to the directory in which the BRM wallet is located.

10. In the following screens, provide the requested information, and then 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.

 **Note:**

During installation, ensure that you enter the existing BRM root password that you provided during the BRM 7.5 installation.

11. In the Installation Summary screen, review your selections, and then click **Install**.

The Installation Progress screen appears, and the installation begins.

 **Note:**

After the installation begins, you cannot stop or cancel the installation.

12. When the installation is done, click **Next**.

The Installation Complete screen appears.

The installer checks for all required software and displays errors if it detects any missing or unavailable components or if any connectivity issues occur.

The next step you perform depends on your upgrade path. If you are upgrading from:

- From BRM 7.5 to BRM 12.0, BRM 12.0 Patch Set 1, or BRM Patch Set 2, proceed to "[Upgrading Your Data to BRM 12.0](#)".
- From BRM 7.5 to 12.0 Patch Set 3 or a later release and want to use the OZT encryption method, proceed to "[Upgrading Your Data to BRM 12.0](#)".
- From BRM 7.5 to BRM 12.0 Patch Set 3 or a later release and want to retain the AES encryption method, proceed to "[Configuring BRM 12.0 Patch Set 3 or Later to Use the AES Encryption Method](#)".

Configuring BRM 12.0 Patch Set 3 or Later to Use the AES Encryption Method

Perform this task if you are upgrading from BRM 7.5 to BRM 12.0 Patch Set 3 or a later release and you want to continue using the AES encryption method. If you skip this step, the OZT encryption method is used.

 **Note:**

The AES encryption method is supported for backwards compatibility only.

To configure BRM to use the AES encryption method:

1. Make a back up of the CRYPTKEY_T table from your BRM 7.5 database.

 **Note:**

If the upgrade of your database schema fails, copy the backed up CRYPTKEY_T table to your BRM database before running the upgrade script for the second time.

2. Log in to the system running the BRM manager. This is generally the system running the BRM database.
3. Generate an encrypted AES key by running the following command:

```
pin_crypt_app -genkey
```

The output states whether the key was generated successfully and, if so, provides the encrypted AES key.

See "pin_crypt_app" in *BRM Developer's Guide* for more information.

4. Write down the encrypted AES key value or copy it to a text editor. Include the **&aes|** because it is part of the encrypted key.
5. In the *BRM_home/sys/dm_oracle/pin.conf* file, set AES as the encryption method for your data:

```
- crypt aes|Encryption_library "&aes|Encrypted_aes_key"
```

where:

- *Encryption_library* is the path and filename of the AES encryption library (**lib_pin_crypt_aes4dm.so**).
- *Encrypted_aes_key* is the encrypted AES key that you generated in step 3.

6. Stop and restart the DM.
7. In the *BRM_home/setup/pin_setup.values* file, set these entries to **NO**:

```
$SETUP_CREATE_PARTITIONS = "NO";
$ENABLE_PARTITION = "NO";
$SETUP_DROP_ALL_TABLES = "NO";
$SETUP_INIT_DB = "NO";
$CREATE_DATABASE_TABLES = "NO";
```

8. Go to the *BRM_home/setup* directory and run the **pin_setup** script.

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 Database Schema



Note:

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

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. The Oracle DM should point to the BRM 12.0 database in which you imported your BRM 7.5 back up data, as described in "[Importing Your BRM 7.5 Data](#)".

Verify that Oracle DM is configured properly by starting **dm_oracle**. 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



Note:

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

These upgrade scripts run a series of scripts that upgrade BRM 7.5 to BRM 12.0 or to a BRM 12.0 patch set:

- For upgrading to BRM 12.0: **pin_upgrade_12.pl**.
- For upgrading to a BRM 12.0 Patch Set: **pin_upgrade_12psx.pl**, where x is the patch set release number, such as 1 for Patch Set 1.

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

Upgrading 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:

 **Note:**

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 a BRM 12.0 Patch Set:

```
perl pin_upgrade_12psx.pl
```

where *x* is the patch set number, such as 1 for Patch Set 1.

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

 **Note:**

Do not press any button until the command prompt asks for the BRM wallet password. Otherwise, the script will consider that to be 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)

 **Note:**

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 a BRM 12.0 Patch Set:

```
perl pin_upgrade_12psx.pl
```

where x is the patch set number, such as 1 for Patch Set 1.

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/sysl/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 a BRM 12.0 Patch Set:

```
perl pin_upgrade_12psx.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/sysdm_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
```

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

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

- 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
```

- 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.

- 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



Note:

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

To upgrade the Pipeline Manager database schema:

- 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.
- 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.
- 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'} );
```

- Save and close the file.

- Grant the required access to user **pin** on the Pipeline Manager tables and sequences, if you have not already done so. See "Loading the Tailor-Made Stored Procedure" in *BRM Installation Guide* for more information about the Pipeline Manager tables and sequences that you should grant user **pin** access to.
- Go to the `Pipeline_home/setup/scripts` directory and run the following command:

 **Note:**

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 a BRM 12.0 Patch Set:

```
perl pin_upgrade_12psx.pl
```

where *x* is the patch set number, such as 1 for Patch Set 1.

- Go to the `Pipeline_home/database/Oracle/Scripts` directory and run the following commands:

```
sqlplus login@database_alias  
Enter password: password
```

```
SQL> @create_pricing_discountmodel_procedures.plb
```

where:

- login* is the user name for the database schema.
- password* is the database schema user password.
- database_alias* is the BRM database alias of the schema.

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*.

 **Note:**

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

To upgrade Business Operations Center from a previous release to Business Operations Center 12.0 Patch Set 3 or a later release, see "[Upgrading Business Operations Center](#)".

Adding Optional Components

At this point, add the optional components that you plan to use in BRM 12.0 or in a BRM 12.0 patch set. See "Installing Individual BRM Components" in *BRM Installation Guide*.

 **Note:**

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 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 the Account Synchronization Queue in the Database

To verify that the Account Synchronization queue is created in the required (BRM or Pipeline) database:

1. Go to the required directory by running the following command:

```
cd $PIN_HOME/sys/dm_ifw_sync/
```

2. Run the following command:

```
pin_ifw_sync_oracle.pl create -q queue_name -t queue_table -l /@ORACLE_SID
```

where:

- *queue_name* is the Account Synchronization queue name. The default is IFW_SYNC_QUEUE.
- *queue_table* is the Account Synchronization queue table name. The default is IFW_SYNC.
- *ORACLE_SID* is the database alias.

Update the Default Settings for the pin_rel Utility

If you use Rated Event (RE) Loader, ensure that the database configurations settings are updated.

To update the default settings for the **pin_rel** utility:

1. Go to the *BRM_home/apps/pin_rel* directory.
2. Open the **Infranet.properties** file.
3. Check and, if necessary, update the entries for the following:

```
infranet.rel.dbname = DATABASE_NAME  
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 **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 **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), run the AMM installation scripts to upgrade your multischema system.

To upgrade AMM in a multischema system:

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

```
perl pin_amt_install.pl
```

Restoring Customizations



Note:

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 a BRM 12.0 patch set:

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

Creating an Oracle AQ Database Queue



Note:

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, re-create your Oracle AQ database queue before you begin synchronizing pricing data. For more information, 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_homel/sys/dm_aq/aa_queue_names` with the re-created Oracle AQ database queue name.

- Restart DM_AQ services.

Loading the pin_notify File

Note:

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

If your system includes Pipeline Manager, 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:

- Go to the *BRM_home/sys/data/config* directory.
- Run the following command:

```
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.

- 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:

- Copy the following files from the *BRM_home/lib* directory to the *BRM_home/PortalDevKit/source/sys/cm* directory:
 - libcmpin.so**
 - libdmpin.so**
- Copy the **cm.cpp** file from the *BRM_home/source/sys/cm* directory to the *BRM_home/PortalDevKit/source/sys/cm* directory.
- Open the *BRM_home/PortalDevKit/source/samples/env.unix* file in a text editor.
- 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, such as */Linux/x86_64/packages/jdk/jdk1.8.0_241*.

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

Note:

SSL-enabled databases are 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 the `create_procedures` Script

After installing the BRM patch set, 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@ORACLE_SID  
Enter password: password
```

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 a BRM 12.0 patch set, 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@ORACLE_SID  
Enter password: password
```

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_auto_reconnect 1  
- 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.

9

Upgrading Business Operations Center

Learn how to upgrade Oracle Communications Business Operations Center to the latest patch set release.

For a list of supported upgrade paths, see "Supported Business Operations Center Upgrades" in *BRM Compatibility Matrix*.

Topics in this document:

- [About Upgrading Business Operations Center](#)
- [Downloading Business Operations Center](#)
- [Upgrading Business Operations Center to the Latest Patch Set Release](#)
- [Upgrading Your Business Operations Center Database Schema Manually](#)

About Upgrading Business Operations Center

The Business Operations Center installer allows you to do a full installation of the software or to do an upgrade of the software. The installer determines which one to perform based on the *Oracle_home* directory specified during the installation process. If you specify an empty *Oracle_home* directory, the installer automatically performs a full installation. If you specify an existing *Oracle_home* directory, the installer automatically performs an upgrade.

During the upgrade process, the Business Operations Center installer does the following:

- Replaces the existing Business Operations Center application WAR file that is deployed in your WebLogic application server with the latest version.
- Copies your existing **Infranet.properties** file and Oracle wallet to the *Domain_home* directory.
- Adds new entries to your Business Operations Center configuration files, such as **bocui-config.properties** and **bocws-config.properties**, retaining any custom values that were in the files.
- Updates your Business Operations Center wallet, retaining any custom values that were in it.
- Upgrades your Business Operations Center database schema to the latest patch set release.

The Business Operations Center installer can automatically upgrade the database schema for you, or you can choose to upgrade the database schema manually at a later time. See "[Upgrading Your Business Operations Center Database Schema Manually](#)".

Downloading Business Operations Center

You can download Business Operations Center from the following locations:

- For Release 12.0: From the Oracle software delivery website (<https://edelivery.oracle.com>)

- For a 12.0 Patch Set: From the Oracle support website (<https://support.oracle.com>)

Search for and download the **Oracle Communications Billing and Revenue Management 12.0.0.x.0** software, where x is **0** for the 12.0 release, **1** for Patch Set 1, **2** for Patch Set 2, and so on. The package includes the **BusinessOperationsCenter_generic.jar** installer.

Upgrading Business Operations Center to the Latest Patch Set Release

Before you upgrade Business Operations Center, ensure that you have upgraded Oracle WebLogic Server and Oracle Database Server to the latest supported versions. For the list of supported versions, see "Business Operations Center Software Compatibility" in *BRM Compatibility Matrix*.

To upgrade to the latest Business Operations Center 12.0 Patch Set release:

1. Make a backup of your existing Business Operations Center installation. If the upgrade process fails, it does not roll back any changes that it made.
2. Verify that your Java Development Kit is installed correctly, including setting your **JAVA_HOME** environment variable and adding the **\$JAVA_HOME/bin** directory in your path.
3. Copy the required third-party libraries to the *Domain_home\lib* directory, where *Domain_home* is the WebLogic Server domain home directory of the domain in which Business Operations Center is deployed. See "Business Operations Center Software Compatibility" in *BRM Compatibility Matrix* for information about the required third-party libraries.
4. Start the Oracle WebLogic Server basic domain administration server or the managed server on which Business Operations Center is deployed.
5. Run the following command, which launches the Business Operations Center installer:

```
Java_home/bin/java -jar -invPtrLoc OracleInventoryPath
BusinessOperationsCenter_generic.jar
```

where:

- *Java_home* is the directory in which you installed the latest compatible Java version. See "Business Operations Center Software Compatibility" in *BRM Compatibility Matrix*.
- *OracleInventoryPath* is the path to your existing Oracle Inventory location.

The Welcome screen appears.

6. Click **Next**.

The Installation Location screen appears.

7. In **Oracle Home**, enter the full path or browse to your existing *Oracle_home* directory for Business Operations Center. Click **Next**.

 **Caution:**

If you enter a new Oracle Home directory, the installer does a full installation of Business Operations Center.

The Features Sets Selection screen appears.

8. Select **Business Operation Center Components** and then click **Next**.

The Prerequisite Libraries Location screen appears.

9. In the **Prerequisite Libraries** field, enter the full path or browse to the directory in which the **ojdbc8.jar** file is stored. Click **Next**.

The WebLogic Server Details screen appears.

10. Enter the details listed in [Table 9-1](#) for the WebLogic Server domain in which Business Operations Center is deployed:

Table 9-1 WebLogic Server Details

Field	Description
Host Name	The IP address or the host name of the machine on which the WebLogic Server domain is configured.
Port Number	The port number assigned to the WebLogic Server domain administration server.
User Name	The WebLogic Server domain administrator user name.
Password	The password for the WebLogic Server domain administrator user.
WebLogic Home	The path of the directory in which the WebLogic Server software is installed.
Use SSL?	Whether to use SSL (deselect for test installations).

 **Note:**

The Business Operations Center installer will not proceed until it verifies that the information you entered is valid. The domain must be in a **RUNNING** state.

The Database Connection Details screen appears.

11. In [Table 9-2](#), enter the database connection details for your BRM database:

Table 9-2 Database Connection Details

Field	Description
Host Name	The IP address or the host name of the machine on which the BRM database is installed.
Port Number	The port number assigned to the BRM database.
Service Name	The database service name.

Table 9-2 (Cont.) Database Connection Details

Field	Description
User Name	The user name for the Business Operations Center schema.

The BOC Schema Upgrade screen appears.

12. In the **Skip Schema Upgrade** checkbox, specify whether to have the installer upgrade the Business Operations Center database schema for you:
 - **Not selected:** The installer upgrades the database schema for you. This is the default.
 - **Selected:** The installer copies the database schema upgrade scripts to the *Oracle_home/scripts/upgrade* directory so you can upgrade the schema later. See "[Upgrading Your Business Operations Center Database Schema Manually](#)".

The Installation Summary screen appears.

13. Confirm your installation selections and then click **Install**.

The installation proceeds.

14. Click **Finish** to complete the installation.

Upgrading Your Business Operations Center Database Schema Manually

If you chose to skip the automatic upgrade of your Business Operations Center database schema, the installer copied the upgrade scripts and a **README.txt** file that explains how to perform the upgrade to your *Oracle_home/scripts/upgrade* directory.

To upgrade your Business Operations Center database schema manually, go the *Oracle_home/scripts/upgrade* directory and run this command:

```
Java_home/bin/java -cp .:OJDBC_Path/ojdbc8.jar:Oracle_home/lib/BOCActions.jar com.oracle.installer.boc.BOCActions upgrade DatabaseHost DatabasePort ServiceName BOCSchemaUser BOCSchemaPassword Oracle_home
```

where:

- *Java_home* is the directory in which you installed the latest compatible Java version. See "Business Operations Center Software Compatibility" in *BRM Compatibility Matrix*.
- *OJDBC_Path* is the directory in which you installed the Oracle JDBC driver.
- *DatabaseHost* is the IP address or the host name of the machine on which the BRM database is installed.
- *DatabasePort* is the port number assigned to the BRM database.
- *ServiceName* is the database service name.
- *BOCSchemaUser* is the Business Operations Center database schema user name.

- *BOCSchemaPassword* is the password for the Business Operations Center database schema user.

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 Pricing Design Center (PDC).

This part 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

Learn 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).

Topics in this document:

- [About Migrating BRM Pricing Data to PDC With ECE](#)
- [Performing Migration of BRM Pricing Data to PDC With ECE](#)
- [Creating a BackUp of Your BRM Database](#)
- [Setting Up the BRM Development System](#)
- [Migrating the BRM Pricing Data to PDC With RRE and BRE](#)
- [Fixing Errors Found in the Analysis Phase](#)
- [Changes Required After Migration](#)
- [Handling Migration Process Errors](#)
- [Restarting the Systems](#)
- [Testing the Migrated Data](#)
- [Importing the Migrated Data to the Production Systems](#)
- [About Working with the Pricing Data after Migration](#)

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 how to create product offerings and to configure setup components. See *PDC Creating Product Offerings* for more information.

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:

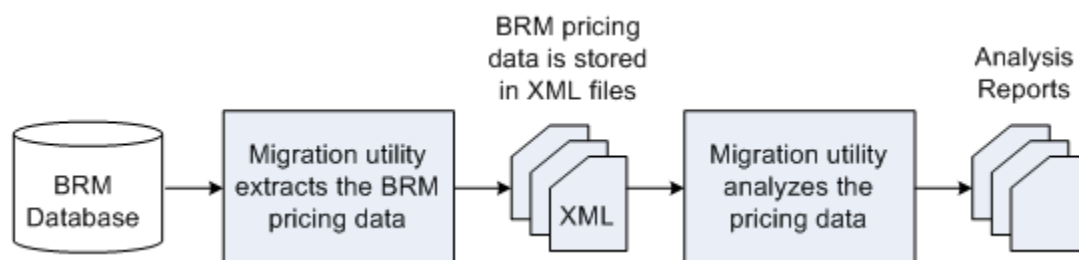
1. The migration utility extracts the BRM pricing data from the BRM database and saves the data in XML files.
2. The migration utility analyzes the data to determine if the configurations are supported in PDC and whether any configurations need to be adjusted prior to the migration.
3. The migration 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 10-1](#) shows the data analysis phase for migrating BRM pricing data to PDC with ECE:

Figure 10-1 Data Analysis Phase for Migrating BRM Pricing Data to PDC With ECE



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.

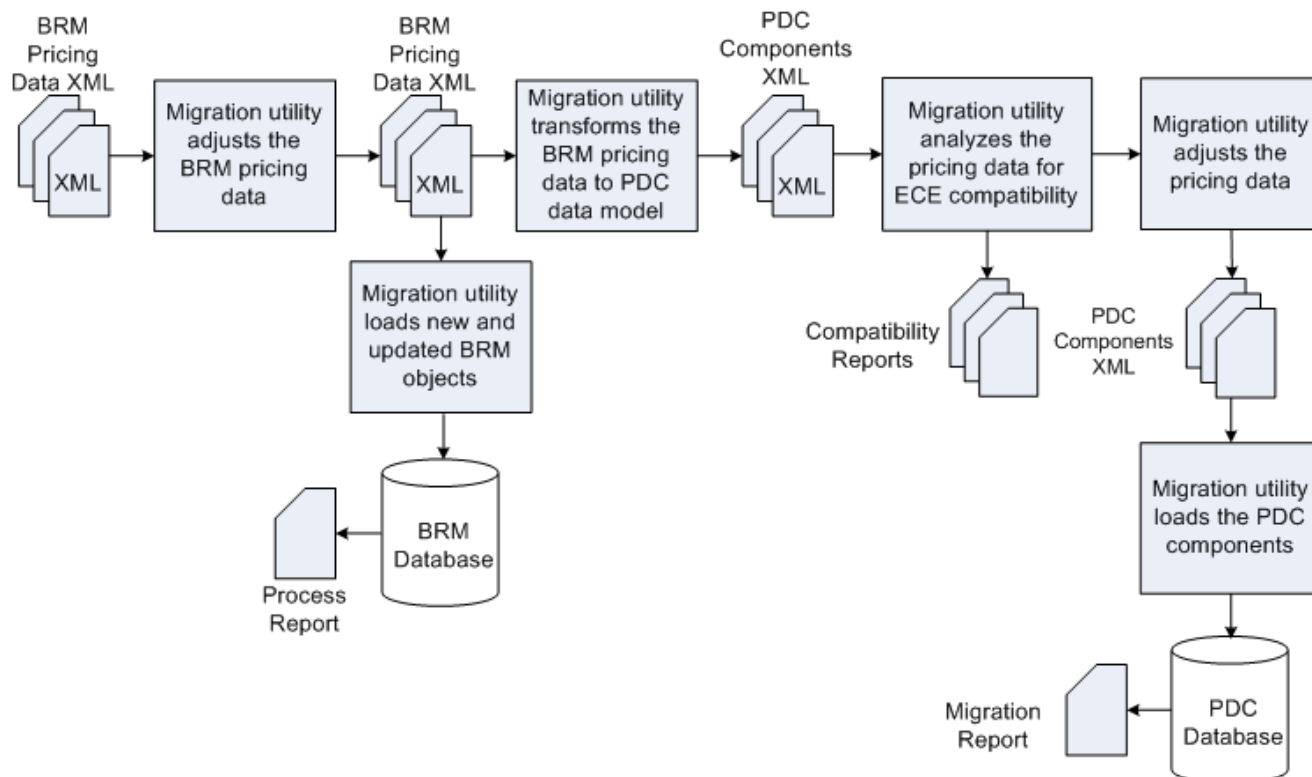
5. The migration utility loads the PDC components into the PDC database.
6. The migration utility loads any new and updated BRM objects (resulting from the data adjustments) into the BRM database.

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 10-2](#) shows the data migration phase for migrating BRM data to PDC with ECE:

Figure 10-2 Data Migration Phase for Migrating BRM Data to PDC With ECE



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 "[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 BRMSee "[Pricing Design Center Pre-Installation Tasks](#)" in *PDC Installation Guide*.
3. Install the complete PDC software.

Note:

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 "Installing Pricing Design Center Software" in *PDC Installation Guide*.

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 **LoadfwConfig**, 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

- 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***lapps/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 PDC with ECE](#)" 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 **LoadfwConfig** 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/LoadfwConfig.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 for PDC with ECE

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

Element	Syntax	Description
walletConfiguration	<pre> <walletConfiguration> <pdwalletLocation>BRMIntegrationPackWallet</pdwalletLocation> </walletConfiguration> </pre>	<p>Specifies the Oracle wallet that stores the sensitive information, such as database passwords, where <i>BRMIntegrationPackWallet</i> is the complete path to the BRM Integration Pack wallet.</p>
xrefDatabase	<pre> <xrefDatabase> <connectionInfo> <login>CrossRefUserName</login> <hostName>CrossRefHostName</hostName> <port>CrossRefPort</port> <serviceName>CrossRefServiceName</serviceName> </connectionInfo> </xrefDatabase> </pre>	<p>Contains the details about the transformation cross-reference database, where:</p> <ul style="list-style-type: none"> • <i>CrossRefUserName</i> specifies the cross-reference database user name • <i>CrossRefHostName</i> specifies the IP address or the host name of the machine on which the cross-reference database is configured • <i>CrossRefPort</i> specifies the port number assigned to the cross-reference database • <i>CrossRefServiceName</i> specifies the name of the cross-reference database service

Table 10-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
migrationDatabase	<pre><migrationDatabase> <connectionInfo> <login>MigrationUserName</login> <hostName>MigrationHostName</hostName> <port>MigrationPort</port> <serviceName>MigrationServiceName</serviceName> </connectionInfo> </migrationDatabase></pre>	<p>Contains the details about the migration cross-reference database, where:</p> <ul style="list-style-type: none"> • <i>MigrationUserName</i> specifies the cross-reference database user name • <i>MigrationHostName</i> specifies the IP address or the host name of the machine on which the cross-reference database is configured • <i>MigrationPort</i> specifies the port number assigned to the cross-reference database • <i>MigrationServiceName</i> specifies the name of the cross-reference database service <p>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.</p>
migrationSource	<pre><migrationSource> <deploymentType>MigrationSourceDeploymentType</deploymentType> <pricingServer> SourcePDCPricingServer </pricingServer> <brmConfiguration> SourceBRMConfiguration </brmConfiguration> </migrationSource></pre>	<p>Contains the details about the migration source system.</p> <ul style="list-style-type: none"> • <deploymentType> element specifies the migration source, where <i>MigrationSourceDeploymentType</i> is: <ul style="list-style-type: none"> 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. • <i>brmConfiguration</i> contains the source BRM system configuration details.

Table 10-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
pricingServer	<pre> <migrationSource> <pricingServer> <connectionInfo> <hostName>SourcePricingServer HostName</hostName> <port>SourcePricingServerPort </port> <adminUser>SourceAdminUserNam e</adminUser> <pdcUser>SourcePDCUser</ pdcUser> <pdcssl>SourceSSLOption</ pdcssl> </connectionInfo> </pricingServer> </migrationSource> </pre>	<p>Contains the source PDC server information, where:</p> <ul style="list-style-type: none"> • <i>SourcePricingServerHostName</i> specifies the IP address or the host name of the machine on which the source PDC server is deployed • <i>SourcePricingServerPort</i> specifies the port number of the domain on which the source PDC server is deployed • <i>SourceAdminUserName</i> specifies the user name of the source PDC server administrator • <i>SourcePDCUser</i> specifies the user name of the source PDC system user • <i>SourceSSLOption</i> specifies whether the source PDC server supports SSL. <p>enabled. Specifies that PDC supports SSL. If SSL is enabled, MigrateBRMPricing uses the t3s://Host:Port URL to access PDC</p> <p>disabled. Specifies that PDC does not support SSL. If SSL is disabled, MigrateBRMPricing uses the t3://Host:Port URL to access PDC</p>

Table 10-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
brmConfiguration	<pre> <migrationSource> <brmConfiguration> <brand>BrandOption</brand> <fieldSelection>EventFieldInfo</ fieldSelection> <breConfig> <containerDesc> <param> <paramname>EDRField</ paramname> <paramvalue>EDRFieldValue </paramvalue> </param> </containerDesc> <eventExtension> <param> <paramname>BREEvent</ paramname> <paramvalue>ExtensionBloc kName</paramvalue> </param> </eventExtension> <serviceExtension> <param> <paramname>BREService</ paramname> <paramvalue>ExtensionBloc kName</paramvalue> </param> </serviceExtension> <defaultValue> <param> <paramname>SERVICE_CLASS< /paramname> <paramvalue>DefaultServic eClass</paramvalue> </param> </defaultValue> </breConfig> </brmConfiguration> </migrationSource> </pre>	<p>Contain the details about the source BRM system configuration.</p> <ul style="list-style-type: none"> • <brand> element specifies whether the source BRM system supports branding, where <i>BrandOption</i> is enabled or disabled. • <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. <pre> <containerDesc> <param> <paramname>EDRC_DESC</paramname> <paramvalue>ALL_RATE</paramvalue> </param> </containerDesc> </pre> • <eventExtension> element specifies to migrate the fields of the specified extension block for the batch rating engine event. <p>Example:</p> <pre> <eventExtension> <param> <paramname>/event/delayed/ session/telco/gsm</paramname> <paramvalue>ASS_GSMW_EXT</ paramvalue> </param> </eventExtension> </pre> • <serviceExtension> element specifies to migrate the fields of the specified extension block for the batch rating engine service. <p>Example:</p> <pre> <serviceExtension> <param> <paramname>/service/telco/gprs</ paramname> <paramvalue>ASS_GPRS_SERV_EXT</ paramvalue> </param> </serviceExtension> </pre> • <defaultValue> element specifies the default service class, where <i>DefaultServiceClass</i> is the default service class name.

Table 10-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
fieldSelection	<pre><fieldSelection> <targetEngine>TargetRatingEngine</ targetEngine> <eventFields> <eventName>EventName</eventName> <fullyQualifiedName>EventFieldName< /fullyQualifiedName> </eventFields> </fieldSelection></pre>	<p>Contains the information about the target rating engine and the BRM event fields provided as input to the target rating engine for usage rating.</p> <p><targetEngine> element specifies the target rating engine used for usage rating, where: <i>TargetRatingEngine</i> is one of the following:</p> <ul style="list-style-type: none"> • 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. <p><eventFields> element specifies the fields that are provided as input to <i>TargetRatingEngine</i>, where:</p> <ul style="list-style-type: none"> • <i>EventName</i> specifies the class name of the BRM event; for example, /event. • <i>EventFieldName</i> specifies the fully qualified name of the BRM event field that is provided as input to <i>TargetRatingEngine</i>; for example, PIN_FLD_NAME. <p>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.</p> <p>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.</p>
migrationTarget	<pre><migrationTarget> <deploymentType>MigrationTargettype</ deploymentType> <pricingServer> TargetPDCPricingServer </pricingServer> <brmConfiguration> TargetBRMConfiguration </brmConfiguration> </migrationTarget></pre>	<p>Contains the details about the migration target system.</p> <ul style="list-style-type: none"> • <deploymentType> element specifies the migration target, where <i>MigrationTargettype</i> is: <ul style="list-style-type: none"> 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 10-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
pricingServer	<pre><migrationTarget> <pricingServer> <connectionInfo> <hostName>TargetPricingServer HostName</hostName> <port>TargetPricingServerPort </port> <adminUser>TargetAdminUserNam e</adminUser> <pdcUser>TargetPDCUser</ pdcUser> <pdcssl>TargetSSLOption</ pdcssl> </connectionInfo> </pricingServer> </migrationTarget></pre>	<p>Contains the target PDC server information, where:</p> <ul style="list-style-type: none"> • <i>TargetPricingServerHostName</i> specifies the IP address or the host name of the machine on which the target PDC server is deployed. • <i>TargetPricingServerPort</i> specifies the port number of the domain on which the target PDC server is deployed. • <i>TargetAdminUserName</i> specifies the user name of the target PDC server administrator. • <i>TargetPDCUser</i> specifies the user name of the target PDC system user. • <i>TargetSSLOption</i> specifies whether the target PDC server supports SSL. <p>enabled specifies that PDC supports SSL. If SSL is enabled, MigrateBRMPricing uses the t3s://Host:Port URL to access PDC</p> <p>disabled specifies that PDC does not support SSL. If SSL is disabled, MigrateBRMPricing uses the t3://Host:Port URL to access PDC</p>
brmConfiguration	<pre><migrationTarget> <brmConfiguration> <loadConfigDir>LoadConfigPath</ loadConfigDir> <loadPriceListDir>LoadPriceList Path</loadPriceListDir> <loadIfwConfigDir>LoadPipelineC onfigPath</loadIfwConfigDir> </brmConfiguration> </migrationTarget></pre>	<p>Contain the details about the target BRM system configuration, where:</p> <ul style="list-style-type: none"> • <i>LoadConfigPath</i> specifies the path to the directory from where the load_config utility is run. • <i>LoadPriceListPath</i> specifies the path to the directory from where the load_price_list utility is run. • <i>LoadPipelineConfigPath</i> specifies the path to the directory from where the LoadIfwConfig utility is run.
logFile	<pre><logFile>MigrationLogFileLocation</ logFile></pre>	<p>Specifies the directory that stores MigrateBRMPricing log files, where <i>MigrationLogFileLocation</i> is the complete path and the name of the log file.</p>
reportFile	<pre><reportFile>ReportFileLocation</ reportFile></pre>	<p>Specifies the directory where MigrateBRMPricing stores the reports generated during the migration process, where <i>ReportFileLocation</i> is the complete path to the directory.</p>
brmExtractedXML	<pre><brmExtractedXML>BRMExtractedXML</ brmExtractedXML></pre>	<p>Specifies the directory where MigrateBRMPricing stores the XML files containing the pricing data extracted from the BRM database, where <i>BRMExtractedXML</i> is the complete path to the directory.</p>

Table 10-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
pdExtractedXML	<code><pdExtractedXML>PDCExtractedXML</pdExtractedXML></code>	Specifies the directory where MigrateBRMPricing stores the XML files containing the pricing data extracted from the PDC database, where <i>PDCExtractedXML</i> is the complete path to the directory.
brmXML	<code><brmXML>BRMDataFile</brmXML></code>	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 <i>BRMDataFile</i> is the complete path to the directory.
pdXML	<code><pdXML>PDCDataFile</pdXML></code>	Specifies the directory where MigrateBRMPricing creates the XML files containing the extracted BRM data in PDC format, where <i>PDCDataFile</i> is the complete path to the directory.
userInputXML	<code><userInputXML>UserInputXML</userInputXML></code>	Specifies the directory that contains the XML user input files that you provide for data adjustments, where <i>UserInputXML</i> is the complete path to the directory.
xrefSQL	<code><xrefSQL>XrefData</xrefSQL></code>	Specifies the directory where MigrateBRMPricing creates the files containing the migration and transformation cross-reference data in sql format, where <i>XrefData</i> is the complete path to the directory.
selectiveMigration	<code><selectiveMigration> <useBatch>MigrationOption</useBatch> <batchName>BatchFileName</batchName></selectiveMigration></code>	Contain the details for selective migration, where: <ul style="list-style-type: none"> <i>MigrationOption</i> is: <ul style="list-style-type: none"> true to specify that selective migration of data must be enabled. false to specify that selective migration of data must be disabled. <i>BatchFileName</i> is the name of the XML file that contains the pricing objects to be migrated.

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:


```
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. Restart the ECE server. See "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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:


```
MigrateBRMPricing -resume
```

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.

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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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:

```
MigrateBRMPricing -pricing -analyze -restart
```

8. Restart the ECE server. See "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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:

```
<!--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:

```
<!--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.

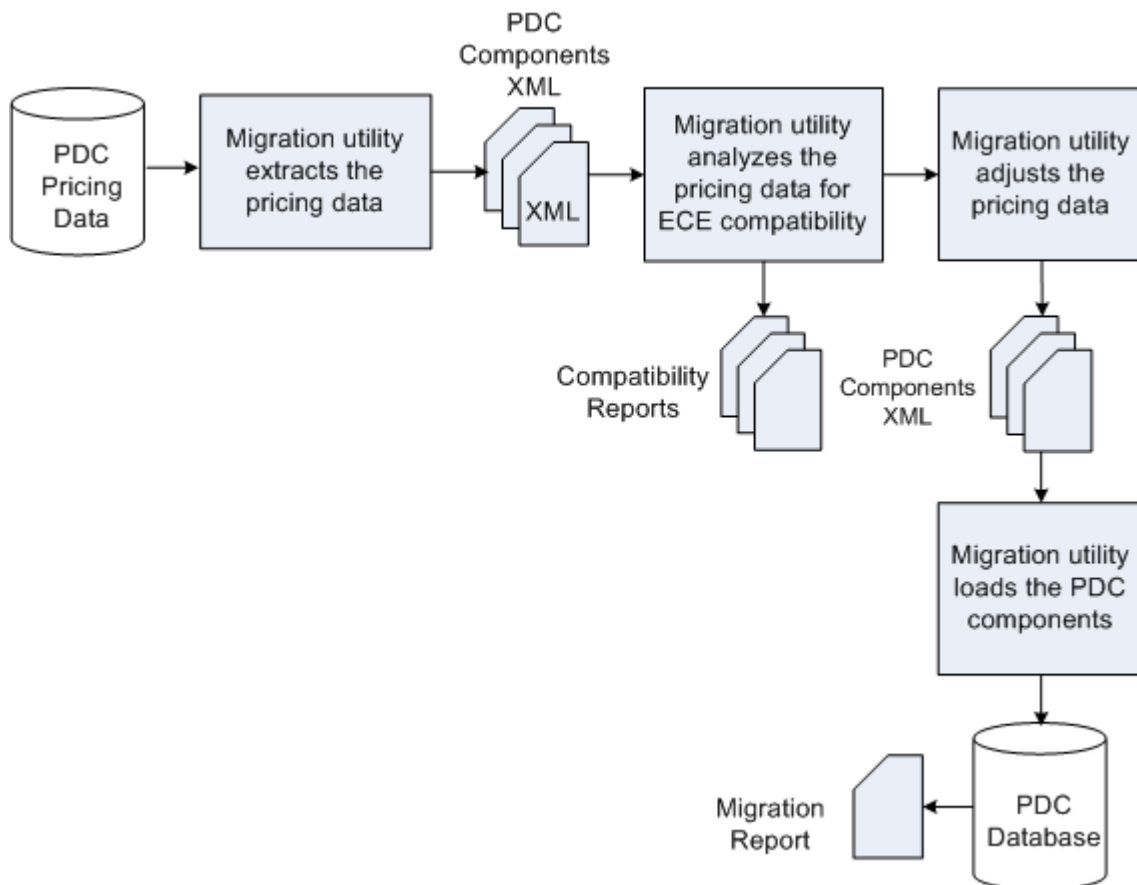
When you restart the migration process, the compatibility reports are also re-generated for the latest data.

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 10-3](#) shows the data migration phase for migrating PDC data to PDC with ECE:

Figure 10-3 Data Migration Phase for Migrating PDC Data to PDC With ECE



For more information on setting up and migrating the PDC pricing data to PDC with ECE, see "[Performing Migration of PDC Pricing 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 set up the PDC development system:

1. Review the system requirements for installing the PDC system.
See "[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 BRMSee "[Pricing Design Center Pre-Installation Tasks](#)" in *PDC Installation Guide*.
3. Install the complete PDC software.

Note:

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.

See "[Installing Pricing Design Center Software](#)" in *PDC Installation Guide*.

4. 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 managed in BRM and their corresponding load utilities in *PDC Creating Product Offerings*.
5. 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.
6. 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.
7. (Optional) Configure the **MigrateBRMPricing** utility. See "[Configuring the Migration Utility for PDC with ECE](#)" for more information.
8. 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.
9. 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](#)").

Note:

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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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:

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 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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
 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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 "Starting and Stopping ECE" in *BRM System Administrator's Guide*.
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 Noncurrency Resources Used in Batch Rating](#)
- [Providing Values List for Expressions in Discount and Chargeshare Configurations](#)

Identifying Resource ID for Noncurrency Resources Used in Batch Rating

If you have a noncurrency 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 example shows the XML that you would provide to map the resource LOYALTY to resource ID 1000019:

```
<?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:

```
<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 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

Rule	Rule Expression	Rate Plan	Impact Category Used in Rate Plan Selector
Rule 1	Attribute values	Rate plan A	IC1
Rule 2	Attribute values	Rate plan A	IC2
Rule 3	Attribute values	Rate plan A	*
Rule 4	Attribute values	Rate plan A	default

Table 10-3 shows the rate objects in rate plan A.

Table 10-3 Rate Plan A

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 1	IC1	\$1.00
Rate Object 1	IC2	\$2.00
Rate Object 1	*	\$5.00
Rate Object 1	default	\$0.50
Rate Object 2	IC1	\$1.50
Rate Object 2	*	\$6.00
Rate Object 3	IC1	\$1.75
Rate Object 3	default	\$0.50
Rate Object 4	default	\$0.50

Solution

Make the changes shown in Table 10-4 to the rate objects:

Table 10-4 Rate Plan Selector

Rule	Rule Expression	Rate Plan	Impact Category Used in Rate Plan Selector
Rule 1	Attribute values	Rateplan A_IC1	IC1
Rule 2	Attribute values	Rateplan A_IC2	IC2
Rule 3	Attribute values	Rateplan A_*	*
Rule 4	Attribute values	Rateplan A_default	default

Table 10-5 shows the rate objects in rate plan A_IC1.

Table 10-5 Rate Plan A_IC1

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 1	IC1	\$1.00
Rate Object 1	IC1 (previously *)	\$5.00
Rate Object 2	IC1	\$1.50
Rate Object 2	IC1 (previously *)	\$5.00
Rate Object 3	IC1	\$0.50
Rate Object 4	IC1 (previously default)	\$1.00

Table 10-6 shows the rate objects in rate plan A_IC2.

Table 10-6 Rate Plan A_IC2

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 1	IC2	\$2.00
Rate Object 1	IC2 (previously *)	\$5.00
Rate Object 2	IC2 (previously *)	\$6.00
Rate Object 3	IC2 (previously default)	\$0.50
Rate Object 4	IC2 (previously default)	\$1.00

Table 10-7 shows the rate objects in rate plan A_IC3.

Table 10-7 Rate Plan A_IC3

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 1	*	\$5.00
Rate Object 2	*	\$6.00
Rate Object 3	*(previously default)	\$0.50

Table 10-7 (Cont.) Rate Plan A_IC3

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 4	* (previously default)	\$0.50

Table 10-8 shows the rate objects in rate plan A_default.

Table 10-8 Rate Plan A_Default

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 1	default	\$0.50
Rate Object 1	default (previously *)	\$5.00
Rate Object 2	default (previously *)	\$6.00
Rate Object 3	default	\$0.50
Rate Object 4	default	\$0.50

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

Origin	Destination	Valid From	Service	Zone Impact Category
123	456	1999-01-01	SMS1	IC1
123	456	1999-01-01	SMS2	IC2

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 noncurrency 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 noncurrency 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 noncurrency 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*.



Note:

After creating the JAR file, restart the Connection Manager (CM) before running the migration utility.

Fixing 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 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, see "[MigrateBRMPricing](#)".

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 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 version="1.0" encoding="UTF-8" ?>
<FieldNameMappings>
  <FieldNameMap>
```

```

    <SourceField>AnyEvent.ACCESS_POINT_NAME</SourceField>
    <TargetField>AnyEvent.ACCESS_POINT_NAME</TargetField>
    <FieldType>EventAttributeSpec</FieldType>
  </FieldNameMap>
</FieldNameMap>
<FieldNameMap>
  <SourceField>EventDelayedSessionTelcoGsm.A_NUMBER</SourceField>
  <TargetField>EventDelayedSessionTelcoGsm.USER_IDENTITY</TargetField>
  <FieldType>EventAttributeSpec</FieldType>
</FieldNameMap>
<FieldNameMap>
  <SourceField>AnyEvent.SERVICE_CLASS</SourceField>
  <TargetField>AnyEvent.SERVICE_CLASS</TargetField>
  <FieldType>EventAttributeSpec</FieldType>
</FieldNameMap>
<FieldNameMap>
  <SourceField>AnyEvent.USAGE_CLASS</SourceField>
  <TargetField>AnyEvent.USAGE_CLASS</TargetField>
  <FieldType>EventAttributeSpec</FieldType>
</FieldNameMap>
<FieldNameMap>
  <SourceField>AnyEvent.RETAIL_ZONE</SourceField>
  <TargetField>AnyEvent.RETAIL_ZONE</TargetField>
  <FieldType>EventAttributeSpec</FieldType>
</FieldNameMap>
<FieldNameMap>
  <SourceField>AnyEvent.SERVICE_CODE</SourceField>
  <TargetField>AnyEvent.SERVICE_CODE</TargetField>
  <FieldType>EventAttributeSpec</FieldType>
</FieldNameMap>
<FieldNameMap>
  <SourceField>EventDelayedSessionTelcoGsm.B_NUMBER</SourceField>
  <TargetField>EventDelayedSessionTelcoGsm.CALLED_ID</TargetField>
  <FieldType>EventAttributeSpec</FieldType>
</FieldNameMap>
</FieldNameMappings>

```

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 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 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>
      </rumExpression>
      <rumName>Duration_batch</rumName>
    </rumSpec>
    <rumSpec>
      <rumExpression>
        <eventFieldExpression>
          <fieldName>EventDelayedSessionTelcoGsm.WHOLESALE_AMOUNT</fieldName>
        </eventFieldExpression>
      </rumExpression>
      <rumName>Passthrough</rumName>
    </rumSpec>
  </eventRUMSpec>
  <productSpecName>TelcoGsmData</productSpecName>
</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](#)
- [Noncurrency 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.

Noncurrency 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 noncurrency balance element is impacted with a positive value, as in a counter, it is migrated as follows:

- If the **Grantable** flag is not checked in the balance impact, it is migrated to a debit in PDC.
- If the **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 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.

Note:

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:
 - i. Search for the **jvmopts** entry.
 - ii. Set the **-Djavax.net.ssl.trustStore** entry to the location of the **client.jks** file copied from Target server.
 - iii. 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 PDC with ECE" for more information:
 - i. Search for **walletConfiguration**.
 - ii. Set the **pdwalletLocation** entry to the location of the BRM Integration Pack wallet copied from the target PDC production system.
 - iii. Search for **pricingServer**.
 - iv. Edit the **connectionInfo** entry within the **pricingServer** element to reference the PDC production system.
 - v. Search for the **xrefDatabase** element.


```
<xrefDatabase>
  <connectionInfo>
    <login>CrossRefUserName</login>
    <hostName>CrossRefHostName</hostName>
    <port>CrossRefPort</port>
    <serviceName>CrossRefServiceName</serviceName>
  </connectionInfo>
</xrefDatabase>
```
 - vi. Edit the **connectionInfo** within the **xrefDatabase** element to reference the PDC transformation database.
 - vii. 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 **pdXML** element:

```
<pdXML>PDCDataFile</pdXML>
```

- 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 managed 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.

Migrating Pricing Data to PDC With RRE and BRE

Learn how to set up your environment and run the 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).

Topics in this document:

- [About Migrating BRM Pricing Data to PDC With RRE and BRE](#)
- [Performing Migration of BRM Pricing Data to PDC With RRE and BRE](#)
- [Creating a BackUp of Your BRM Database](#)
- [Setting Up the BRM Development System](#)
- [Migrating the BRM Pricing Data to PDC With RRE and BRE](#)
- [Fixing Errors Found in the Analysis Phase](#)
- [Changes Required After Migration](#)
- [Handling Migration Process Errors](#)
- [Restarting the Systems](#)
- [Testing the Migrated Data](#)
- [Importing the Migrated Data to the Production Systems](#)
- [About Working with the Pricing Data after Migration](#)

Pricing data migration should be performed by experienced pricing administrators. You should have a basic understanding of how to create product offerings and configure setup components in PDC. See *PDC Creating Product Offerings* for more information.

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, **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. 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:

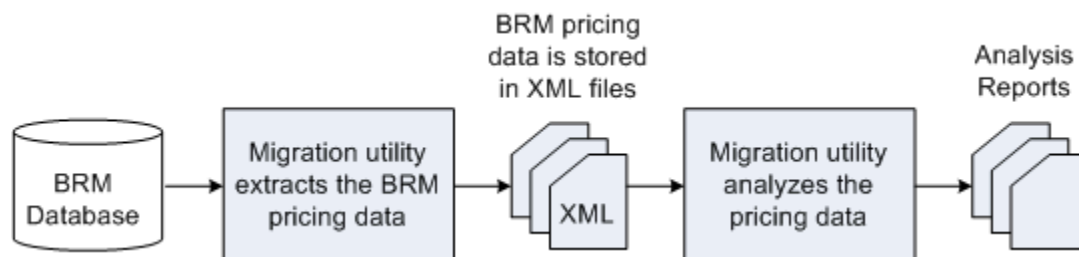
1. The migration utility extracts the BRM pricing data from the BRM database and saves the data in XML files.
2. The migration utility analyzes the data to determine if the configurations are supported in PDC and whether any configurations need to be adjusted prior to the migration.
3. The migration 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 11-1 shows the data analysis phase:

Figure 11-1 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

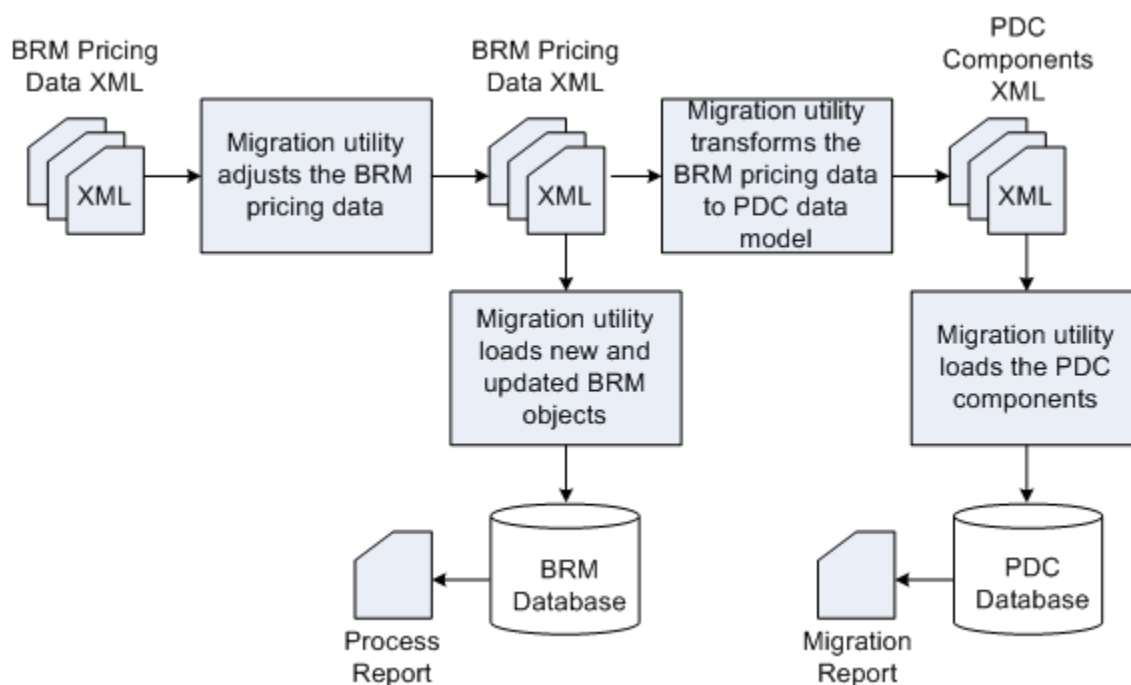
The data migration phase is the second phase of the migration process. You must *resume* the migration process after the analysis phase is complete to start the data migration.

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 11-2](#) shows the data migration phase:

Figure 11-2 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 BRMSee the discussion about Pricing Design Center pre-Installation tasks in *PDC Installation Guide*.
3. Install the complete PDC software.

Note:

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 one of the following migration options:

- **Migrate RRE and BRE pricing data to PDC**

See the discussion about installing Pricing Design Center complete software in *PDC Installation 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 **LoadfwConfig**, 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

- 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 PDC with RRE and BRE](#)" 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 **LoadfwConfig** 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\LoadfwConfig.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>
```

```
</xs:key>
</xs:element>
```

10. Obtain the Java Keystore password.

Configuring the Migration Utility for PDC with RRE and BRE

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 11-1](#) lists the elements in **MigrateConfiguration.xml** and the syntax and description for each element.

Table 11-1 Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
walletConfiguration	<pre><walletConfiguration> <pdwalletLocation>BRMIntegrationPackWallet</pdwalletLocation> </walletConfiguration></pre>	Specifies the Oracle wallet that stores the sensitive information, such as database passwords, where <i>BRMIntegrationPackWallet</i> is the complete path to the BRM Integration Pack wallet.
xrefDatabase	<pre><xrefDatabase> <connectionInfo> <login>CrossRefUserName</login> <hostName>CrossRefHostName</hostName> <port>CrossRefPort</port> <serviceName>CrossRefServiceName</serviceName> </connectionInfo> </xrefDatabase></pre>	<p>Contains the details about the transformation cross-reference database, where:</p> <ul style="list-style-type: none"> • <i>CrossRefUserName</i> specifies the cross-reference database user name • <i>CrossRefHostName</i> specifies the IP address or the host name of the machine on which the cross-reference database is configured • <i>CrossRefPort</i> specifies the port number assigned to the cross-reference database • <i>CrossRefServiceName</i> specifies the name of the cross-reference database service

Table 11-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
migrationDatabase	<pre><migrationDatabase> <connectionInfo> <login>MigrationUserName</login> <hostName>MigrationHostName</hostName> <port>MigrationPort</port> <serviceName>MigrationServiceName</serviceName> </connectionInfo> </MigrationDatabase></pre>	<p>Contains the details about the migration cross-reference database, where:</p> <ul style="list-style-type: none"> • <i>MigrationUserName</i> specifies the cross-reference database user name • <i>MigrationHostName</i> specifies the IP address or the host name of the machine on which the cross-reference database is configured • <i>MigrationPort</i> specifies the port number assigned to the cross-reference database • <i>MigrationServiceName</i> specifies the name of the cross-reference database service <p>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.</p>
migrationSource	<pre><migrationSource> <deploymentType>MigrationSourcetype</deploymentType> <pricingServer> SourcePDCPricingServer </pricingServer> <brmConfiguration> SourceBRMConfiguration </brmConfiguration> </migrationSource></pre>	<p>Contains the details about the migration source system.</p> <ul style="list-style-type: none"> • <deploymentType> element specifies the migration source, where <i>MigrationSourcetype</i> 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.

Table 11-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
pricingServer	<pre> <migrationSource> <pricingServer> <connectionInfo> <hostName>SourcePricingServer HostName</hostName> <port>SourcePricingServerPort </port> <adminUser>SourceAdminUserNam e</adminUser> <pdcUser>SourcePDCUser</ pdcUser> <pdcssl>SourceSSLOption</ pdcssl> </connectionInfo> </pricingServer> </migrationSource> </pre>	<p>Contains the source PDC server information, where:</p> <ul style="list-style-type: none"> • <i>SourcePricingServerHostName</i> specifies the IP address or the host name of the machine on which the source PDC server is deployed • <i>SourcePricingServerPort</i> specifies the port number of the domain on which the source PDC server is deployed • <i>SourceAdminUserName</i> specifies the user name of the source PDC server administrator • <i>SourcePDCUser</i> specifies the user name of the source PDC system user • <i>SourceSSLOption</i> specifies whether the source PDC server supports SSL. <p>enabled specifies that PDC supports SSL. If SSL is enabled, MigrateBRMPricing uses the t3s://Host:Port URL to access PDC</p> <p>disabled specifies that PDC does not support SSL. If SSL is disabled, MigrateBRMPricing uses the t3://Host:Port URL to access PDC</p>

Table 11-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
brmConfiguration	<pre> <migrationSource> <brmConfiguration> <brand>BrandOption</brand> <skipBREMigration>SkipOption</ skipBREMigration> <fieldSelection>EventFieldInfo</ fieldSelection> <breConfig> <containerDesc> <param> <paramname>EDRField</ paramname> <paramvalue>EDRFieldValue </paramvalue> </param> </containerDesc> <eventExtension> <param> <paramname>BREEvent</ paramname> <paramvalue>ExtensionBloc kName</paramvalue> </param> </eventExtension> <serviceExtension> <param> <paramname>BREService</ paramname> <paramvalue>ExtensionBloc kName</paramvalue> </param> </serviceExtension> <defaultValue> <param> <paramname>SERVICE_CLASS< /paramname> <paramvalue>DefaultServic eClass</paramvalue> </param> </defaultValue> </breConfig> </brmConfiguration> /<migrationSource> </pre>	<p>Contain the details about the source BRM system configuration.</p> <ul style="list-style-type: none"> • <brand> element specifies whether the source BRM system supports branding, where <i>BrandOption</i> is: <ul style="list-style-type: none"> enabled to specify that the source BRM system supports branding disabled to specify that the source BRM system does not support branding • <skipBREMigration> element specifies whether to skip migration of pipeline configuration data, where <i>SkipOption</i> is either true or false. The default is false for PDC with BRE and RRE. • <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. <pre> <containerDesc> <param> <paramname>EDRC_DESC</paramname> <paramvalue>ALL_RATE</paramvalue> </param> </containerDesc> </pre> • <eventExtension> element specifies to migrate the fields of the specified extension block for the batch rating engine event. Example: <pre> <eventExtension> <param> <paramname>/event/delayed/ session/telco/gsm</paramname> <paramvalue>ASS_GSMW_EXT</ paramvalue> </param> </eventExtension> </pre> • <serviceExtension> element specifies to migrate the fields of the specified extension block for the batch rating engine service. Example: <pre> <serviceExtension> <param> <paramname>/service/telco/gprs</ paramname> </pre>

Table 11-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
		<pre> <paramvalue>ASS_GPRS_SERV_EXT</paramvalue> </param> </serviceExtension> </pre> <ul style="list-style-type: none"> • <defaultValue> element specifies the default service class, where <i>DefaultServiceClass</i> is the default service class name.
fieldSelection	<pre> <fieldSelection> <targetEngine>TargetRatingEngine</targetEngine> <eventFields> <eventName>EventName</eventName> <fullyQualifiedName>EventFieldName</fullyQualifiedName> </eventFields> </fieldSelection> </pre>	<p>Contains the information about the target rating engine and the BRM event fields provided as input to the target rating engine for usage rating.</p> <p><targetEngine> element specifies the target rating engine used for usage rating, where: <i>TargetRatingEngine</i> is one of the following:</p> <ul style="list-style-type: none"> • 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. <p><eventFields> element specifies the fields that are provided as input to <i>TargetRatingEngine</i>, where:</p> <ul style="list-style-type: none"> • <i>EventName</i> specifies the class name of the BRM event; for example, /event. • <i>EventFieldName</i> specifies the fully qualified name of the BRM event field that is provided as input to <i>TargetRatingEngine</i>; for example, PIN_FLD_NAME. <p>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.</p> <p>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.</p>
migrationTarget	<pre> <migrationTarget> <deploymentType>MigrationTargettype</deploymentType> <pricingServer> TargetPDCPricingServer </pricingServer> <brmConfiguration> TargetBRMConfiguration </brmConfiguration> </migrationTarget> </pre>	<p>Contains the details about the migration target system.</p> <ul style="list-style-type: none"> • <deploymentType> element specifies the migration target, where <i>MigrationTargettype</i> 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. • <pricingServer> element contains the target PDC server information • <brmConfiguration> element contains the target BRM system configuration details.

Table 11-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
pricingServer	<pre><migrationTarget> <pricingServer> <connectionInfo> <hostName>TargetPricingServer HostName</hostName> <port>TargetPricingServerPort </port> <adminUser>TargetAdminUserNam e</adminUser> <pdidUser>TargetPDCUser</ pdidUser> <pdcssl>TargetSSLOption</ pdcssl> </connectionInfo> </pricingServer> </migrationTarget></pre>	<p>Contains the target PDC server information, where:</p> <ul style="list-style-type: none"> • <i>TargetPricingServerHostName</i> specifies the IP address or the host name of the machine on which the target PDC server is deployed • <i>TargetPricingServerPort</i> specifies the port number of the domain on which the target PDC server is deployed • <i>TargetAdminUserName</i> specifies the user name of the target PDC server administrator • <i>TargetPDCUser</i> specifies the user name of the target PDC system user • <i>TargetSSLOption</i> specifies whether the target PDC server supports SSL. <p>enabled specifies that PDC supports SSL. If SSL is enabled, MigrateBRMPricing uses the t3s://Host:Port URL to access PDC</p> <p>disabled specifies that PDC does not support SSL. If SSL is disabled, MigrateBRMPricing uses the t3://Host:Port URL to access PDC</p>
brmConfiguration	<pre><migrationTarget> <brmConfiguration> <loadConfigDir>LoadConfigData</ loadConfigDir> <loadPriceListDir>LoadPriceData </loadPriceListDir> <loadIfwConfigDir>LoadPipelineC onfigData<loadIfwConfigDir> </brmConfiguration> </migrationTarget></pre>	<p>Contain the details about the target BRM system configuration, where:</p> <ul style="list-style-type: none"> • <i>LoadConfigPath</i> specifies the path to the directory from where the load_config utility is run. • <i>LoadPriceListPath</i> specifies the path to the directory from where the load_price_list utility is run. • <i>LoadPipelineConfigPath</i> specifies the path to the directory from where the LoadIfwConfig utility is run.
logFile	<pre><logFile>MigrationLogFileLocation</ logFile></pre>	<p>Specifies the directory that stores MigrateBRMPricing log files, where <i>MigrationLogFileLocation</i> is the complete path and the name of the log file.</p>
reportFile	<pre><reportFile>ReportFileLocation</ reportFile></pre>	<p>Specifies the directory where MigrateBRMPricing stores the reports generated during the migration process, where <i>ReportFileLocation</i> is the complete path to the directory.</p>
brmExtractedXML	<pre><brmExtractedXML>BRMExtractedXML</ brmExtractedXML></pre>	<p>Specifies the directory where MigrateBRMPricing stores the XML files containing the pricing data extracted from the BRM database, where <i>BRMExtractedXML</i> is the complete path to the directory.</p>

Table 11-1 (Cont.) Elements in the MigrateConfiguration.xml File

Element	Syntax	Description
pdExtractedXML	<pdExtractedXML>PDCExtractedXML</pdExtractedXML>	Specifies the directory where MigrateBRMPricing stores the XML files containing the pricing data extracted from the PDC database, where <i>PDCExtractedXML</i> is the complete path to the directory.
brmXML	<brmXML>BRMDataFile</brmXML>	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 <i>BRMDataFile</i> is the complete path to the directory.
pdXML	<pdXML>PDCDataFile</pdXML>	Specifies the directory where MigrateBRMPricing creates the XML files containing the extracted BRM data in PDC format, where <i>PDCDataFile</i> is the complete path to the directory.
userInputXML	<userInputXML>UserInputXML</userInputXML>	Specifies the directory that contains the XML user input files that you provide for data adjustments, where <i>UserInputXML</i> is the complete path to the directory.
xrefSQL	<xrefSQL>XrefData</xrefSQL>	Specifies the directory where MigrateBRMPricing creates the files containing the migration and transformation cross-reference data in sql format, where <i>XrefData</i> is the complete path to the directory.

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](#)".

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
```

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.

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
```

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 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
```

8. Repeat steps 6 and 7 until there are no errors in the pricing analysis reports.
9. 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.

10. Review the migration report and verify that all pricing objects were migrated to PDC successfully.
11. 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](#)".
 - 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
```

 **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.

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
```

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 Noncurrency Resources Used in Batch Rating

If you have a noncurrency 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 example shows the XML that you would provide to map the resource LOYALTY to resource ID 1000019.

```
<?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:

```
<Filters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="\oracle\communications\brm\pdc\migration\pricing\impl\br
e\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 11-2 shows a rate plan selector with multiple rules that result in the same rate plan (rate plan A) with different impact categories:

Table 11-2 Rate Plan Selector

Rule	Rule Expression	Rate Plan	Impact Category Used in Rate Plan Selector
Rule 1	Attribute values	Rate plan A	IC1
Rule 2	Attribute values	Rate plan A	IC2
Rule 3	Attribute values	Rate plan A	*
Rule 4	Attribute values	Rate plan A	default

Table 11-3 shows the rate objects in rate plan A.

Table 11-3 Rate Plan A

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 1	IC1	\$1.00
Rate Object 1	IC2	\$2.00
Rate Object 1	*	\$5.00
Rate Object 1	default	\$0.50
Rate Object 2	IC1	\$1.50
Rate Object 2	*	\$6.00
Rate Object 3	IC1	\$1.75
Rate Object 3	default	\$0.50
Rate Object 4	default	\$0.50

Solution

Make the changes shown in Table 11-4 to the rate objects:

Table 11-4 Rate Plan Selector

Rule	Rule Expression	Rate Plan	Impact Category Used in Rate Plan Selector
Rule 1	Attribute values	Rateplan A_IC1	IC1
Rule 2	Attribute values	Rateplan A_IC2	IC2
Rule 3	Attribute values	Rateplan A_*	*
Rule 4	Attribute values	Rateplan A_default	default

Table 11-5 shows the rate objects in rate plan A_IC1.

Table 11-5 Rate Plan A_IC1

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 1	IC1	\$1.00
Rate Object 1	IC1 (previously *)	\$5.00
Rate Object 2	IC1	\$1.50
Rate Object 2	IC1 (previously *)	\$5.00
Rate Object 3	IC1	\$0.50
Rate Object 4	IC1 (previously default)	\$1.00

Table 11-6 shows the rate objects in rate plan A_IC2.

Table 11-6 Rate Plan A_IC2

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 1	IC2	\$2.00
Rate Object 1	IC2 (previously *)	\$5.00
Rate Object 2	IC2 (previously *)	\$6.00
Rate Object 3	IC2 (previously default)	\$0.50
Rate Object 4	IC2 (previously default)	\$1.00

Table 11-7 shows the rate objects in rate plan A_IC3.

Table 11-7 Rate Plan A_IC3

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 1	*	\$5.00
Rate Object 2	*	\$6.00
Rate Object 3	* (previously default)	\$0.50
Rate Object 4	* (previously default)	\$0.50

Table 11-8 shows the rate objects in rate plan A_default.

Table 11-8 Rate Plan A_Default

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 1	default	\$0.50
Rate Object 1	default (previously *)	\$5.00
Rate Object 2	default (previously *)	\$6.00
Rate Object 3	default	\$0.50

Table 11-8 (Cont.) Rate Plan A_Default

Rate Object	Impact Category Used in Rate Object	Balance Impact
Rate Object 4	default	\$0.50

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

Origin	Destination	Valid From	Service	Zone Impact Category
123	456	1999-01-01	SMS1	IC1
123	456	1999-01-01	SMS2	IC2

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 noncurrency 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 noncurrency 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 noncurrency 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*.



Note:

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](#)
- [Noncurrency 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.

Noncurrency 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 noncurrency balance element is impacted with a positive value, as in a counter, it is migrated as follows:

- If the **Grantable** flag is not checked in the balance impact, it is migrated to a debit in PDC.
- If the **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.

Note:

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:
 - a. 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.
 - b. Propagate any updates that you made to the BRM data in the development system *during* the original migration to the BRM production system.
 - c. 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.
 - 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 **LoadfwConfig** 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 **LoadfwConfig** utility for each Pipeline Manager production database.
For example, **LoadfwConfig_hostname**, where *hostname* is the name of the Pipeline Manager production database server.
 - e. Copy the contents from the **LoadfwConfig** directory installed on the production system (the default directory is **XmlLoader**).
 - f. Ensure that the **LoadfwConfig** 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:
 - i. Search for the **jvmopts** entry.
 - ii. Set the **-Djavax.net.ssl.trustStore** entry to the location of the **client.jks** file copied from Target server.
 - iii. 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_MigrationConfiguration.xml*, where *Hostname* identifies the target PDC system.
 - b. Edit *Hostname_MigrateConfiguration.xml* to reference the target production systems.
 - i. Search for **walletConfiguration**.
 - ii. Set the **pdcWalletLocation** entry to the location of the BRM Integration Pack wallet copied from the target PDC production system.
 - iii. Search for **pricingServer**:

```

<pricingServer>
  <connectionInfo>
    <hostName>PricingServerHostName</hostName>
    <port>PricingServerPort</port>
    <adminUser>AdminUserName</adminUser>
    <pdidUser>PDCUser</pdidUser>
  </connectionInfo>
</pricingServer>

```

- iv. Edit the **connectionInfo** entry within the **pricingServer** element to reference the PDC production system.
- v. Search for the **xrefDatabase** element.

```
<xrefDatabase>
  <connectionInfo>
    <login>CrossRefUserName</login>
    <hostName>CrossRefHostName</hostName>
    <port>CrossRefPort</port>
    <serviceName>CrossRefServiceName</serviceName>
  </connectionInfo>
</xrefDatabase>
```

- vi. Edit the **connectionInfo** within the **xrefDatabase** element to reference the PDC transformation database.
- vii. Search for the **targetBRM** element.
- viii. Edit the **loadPriceListDir** within the **targetBRM** element to reference the directory where **loadpricelist** utility is run to update the BRM system.
- ix. (Only if Pipeline data was migrated by the original migration) Edit the **loadIrwConfigDir** within the **targetBRM** element to reference the directory where **LoadIrwConfig** 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:

```
<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 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 managed 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.

12

Migration Reports

Learn about the Oracle Communications Pricing Design Center (PDC) Migration Reports.

Topics in this document:

- [About 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 12-1 shows a sample product analysis output for the rate plan object.

Figure 12-1 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

UserInput - User has to provide input to resolve the conflicts

BRM Code	BRM Name	Analysis Message	Severity Type
RADJSDLMS	Rate adjust and special day link for multiple services	The charge rate plan, "RADJSDLMS", has configurations for the service codes, "SMS", which are not used in any product. These configurations will not be migrated.	Warning
-	-	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.	Critical

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.

13

BRM Pricing Migration Utility

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

Topics in this document:

- [MigrateBRMPricing](#)

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 MigrateConfigurtion_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.

 **Note:**

You can use the **-skip** parameter only if you are migrating pricing objects to PDC with ECE. You cannot use it with the **-config** parameter. 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.

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.