Oracle® Communications Billing and Revenue Management

Installation Guide

Release 15.2
G35884-01
January 2026

Oracle Communications Billing and Revenue Management Installation Guide, Release 15.2

G35884-01

Contents

About This Content

Part I Preparing for BRM Installation

1 BRM Installation Overview

About Planning and Installing a BRM System 1
Types of BRM Systems 2
BRM Demonstration or Development System 2
BRM Production System 2
Small Production System 3
Mid-Size Production System 4
High-Availability Production System 5
A BRM Multischema Production System 5
Installation Overview 8
Installing and Configuring a Localized Version of BRM 9
Selecting the Database Character Encoding 9

2 Planning Your Database Configuration

Configuring Development or Demonstration System Databases 1
Tablespace Mapping 1
Creating Tablespaces 1
Configuring init.ora Parameters for Development or Demonstration Systems 1
Configuring Production System Databases 2
Storage Considerations, Redo, and Undo 2
Estimating the Database Size 3
International Version Sizing Considerations 4
Selecting the Storage Model 4
Creating Tablespace Storage 5
Creating Tablespaces 6
Running Configuration Scripts to Create the BRM Database 6
Configuring init.ora Parameters 6

Using Optimization 7
About Using Virtual Columns 8

3 System Requirements

Overview of Hardware and Software Requirements 1
Operating System Requirements 1
About Critical Patch Updates 2
Database Requirements 2
Network Requirements 2
TCP/IP 2
IP Addresses 2
Assigning Disks for the Operating System and for BRM 3
About Network Performance 3
About Oracle Exalogic and Oracle Exadata 3
Additional Requirements 4
Information Requirements for BRM Installation 4
Information Requirements for All BRM Installations 4
BRM File Location Details 4
BRM Oracle Wallet Details 5
BRM Database SSL Details 5
BRM Database System User Details 6
BRM Database Connection Details 6
BRM Connection Manager Details 6
Enterprise Application Integration Framework Details 7
Oracle Data Manager Details 7
BRM Table and Index Tablespace Details 7
Additional Information Requirements for BRM Complete Installation 7
Oracle Connection Manager Proxy Details 8
Oracle Connection Manager Master Process Details 8
Pipeline Schema User Details 8
BRM Schema User Details 8

Part II Installing BRM

4 BRM Preinstallation Tasks

Installing the Database 1
Collecting Information Before Installing the Oracle Database Software 2
About Capturing Information When Installing the Oracle Database Software 2
Before Installing Oracle Database Software 2
Installing and Configuring Your Oracle Database: Task List 2

Installing the Oracle Database Software 3
Creating your BRM Database 3
Setting Environment Variables for the Database 4
Setting Up Your Database for BRM 4
Setting Up Oracle RAC 6
Installing the Database and Oracle DM on Separate Machines 8
Modifying Your Oracle Database Installation 9
Installing Third-Party Tools 9
Installing Java 9
Installing Perl 10
Installing Apache Kafka 11
Preparing the Environment 11
Downloading the JDBC Driver 11
Creating and Configuring pin_user 12
Setting the Environment Variables for Installation 13
Setting BRM Wallet Location in sqlnet.ora 14

5 Installing BRM

About the BRM Installation Package 1
Installing BRM: Task List 3
Downloading the BRM Software 4
Installing BRM in GUI Mode 4
Starting the BRM GUI Installer 4
Installing a Typical BRM System 6
Installing All BRM Components 12
Installing Individual BRM Components 20
Installing CM and DM on Separate Machines 22
Installing BRM in Silent Mode 23
Creating a Response File 23
Performing a Silent Installation 24
Completing the Installation 25

6 BRM Postinstallation Tasks

Scheduling Backups 1
Referencing Environment Variables in pin.conf Files 1
Postinstallation Tasks for a BRM Non-Production System 2
Editing the pin_setup.values File 2
Running the pin_setup Script 4
Changing Your Database Partitions 5
Setting Environment Variables for Pipeline Manager 5

Updating Client Wallet for Pipeline Manager 6
Copying Files for Compiling CM and FM Modules 6
Editing and Loading the Event Notification List 6
Loading the Tailor-Made Stored Procedure 7
Postinstallation Tasks for a BRM Production Installation 8
Editing the pin_tables.values File 9
Enabling Different Classes for Partitioning during Installation 9
Editing Optional Components Configuration Entries 10
Maintaining a BRM Database 10
Maintaining the Connection to the Database 11
Monitoring Database Space 11
Rebuilding Indexes 12
Configuring Views for Oracle Application Infrastructure Architecture 12
What's Next? 14

7 Installing a Multischema System

Before Installing a Multischema System 1
Multischema Installation Overview 1
Installing a Multischema System: Task List 2
Installing BRM on the Primary Installation Machine 4
Installing BRM on a Secondary Installation Machine 5
Verifying That the Installation Machines Connect to All Schemas 6
Using testnap to Verify Access to Your Schemas 6
Installing and Configuring Multidatabase Manager on the Primary Installation Machine 7
Disabling Creation of Uniqueness Objects 8
Configuring pin_multidb.conf on the Primary Installation Machine 8
Configuring Client Wallet on the Primary Installation Machine 10
Connecting to SSL-Enabled Oracle Database 11
Completing Configuration of the Primary Installation System 12
Completing Configuration of the Secondary Installation System 13
Enforcing the Creation of Unique Account Numbers 14
Setting Up Schema Qualifications 14
Updating Views for Multischema Systems 16
What's Next? 16

8 Installing BRM Thick Clients

About Installing BRM Clients 1
About the BRM Client Package 2
Downloading the BRM Client Software 2
About Installing Self-Care Manager 3

Installing BRM Clients in GUI Mode 3
Installing All BRM Clients 3
Installing Individual BRM Clients 5
Installing BRM Clients in Silent Mode 8
Creating a Response File 8
Performing a Silent Installation 9
Configuring the Localization SDK on Windows 10
Granting Administrative Privileges to Pricing Center Users 10
Modifying the Database Configuration File 10
Modifying the pricing_admin.pl Script Configuration File 11
Setting Up the Database Server for the Pipeline Manager Administrator 13
Initializing the Pricing Admin Configuration Object 13
Specifying Administrative Privileges for Pipeline Manager Users 14
Removing Pricing Admins 14
Starting a BRM Client Application on Linux 15
Starting a BRM Client Application on Windows 15
Problems Installing BRM Client Applications 15
Problem: Cannot Start a Client Application after Installation 16
Possible Cause 16
Solution 16
What's Next? 16

9 Installing Pipeline Manager

Pipeline Manager Installation Overview 1
Pipeline Manager System Requirements 1
Pipeline Manager Preinstallation Tasks 1
Creating a User and Configuring Environment Variables 2
Setting the Maximum Allowed Number of Open Files 3
Configuring an Oracle Pipeline Manager Database 4
Setting the Environment for the Pipeline Manager Database 4
Setting Up the Oracle JSA Database Schema 4
Setting Up the Oracle Pipeline Manager Framework Database Schema 5
Changing Public Synonyms to Private for Users in Multiuser Environments 6
Loading Procedures for FCT_DuplicateCheck 7
Installing Pipeline Manager 7
Pipeline Manager Directory Structure 8
Installing Pipeline Manager Optional Components 9
Increasing Heap Size to Avoid “Out of Memory" Error Messages 9
Uninstalling Pipeline Manager and Optional Components 10

10 Installing Pipeline Configuration Center

PCC Installation Overview 1

About the PCC Architecture 2

Overview of the PCC Installation Procedure 2

Performing a Secure PCC Installation 3

Ensuring a Successful PCC Installation 3

PCC System Requirements 4

PCC Preinstallation Tasks 4

Installing and Configuring Oracle WebLogic Server 4

Installing Oracle WebLogic Server 5

Installing and Configuring Oracle Application Development Runtime 5

Creating an Oracle WebLogic Server Domain 5

Configuring SSL for the Oracle WebLogic Server Domain 6

Setting the JAVA_HOME Environment Variable 6

Downloading the JDBC Driver for PCC 7

Installing and Configuring BRM 7

Setting Up the Business Parameter 7

Downloading the PCC Software 8

Installing PCC 8

Installing PCC in GUI Mode 9

Installing PCC in Silent Mode 13

About the Response File Template 13

Creating a Response File 13

Installing PCC in Silent Mode 14

About Installation Logs 15

PCC Postinstallation Tasks 15

Creating a PCC User after Installing PCC 16

Enabling Authentication Cookies 16

Configuring the Session Timeout 17

Configuring SAML 2.0 for SSO Using a Service Provider 18

Creating a SAML2 Assertion Provider 19

Creating a SAML2 Web Single Sign-On Identity Provider Partner 19

Creating a SAML2 Authenticator 20

Configuring SAML2 on the Administration Server 21

Configuring the SP metadata.xml file for the Identity Provider 21

Updating the Deployment Plan of PCC 22

Verifying the PCC Installation 23

Checking the State of the Installed Components 23

Logging In to PCC 23

About File Permissions 23

Verifying Configuration Entries 24

Uninstalling PCC 24
Uninstalling PCC in Silent Mode 25
Deleting Files after Uninstalling PCC 25
Administering PCC 26
Managing PCC Security 26
Monitoring PCC 26
Managing PCC 26
Troubleshooting PCC 26
Troubleshooting Checklist 27
Using Error Logs to Troubleshoot PCC 27
Diagnosing PCC Problems 27
Refreshing the Browser 27
Unable to Perform Any Task after Logging In 27
Getting Help for PCC Problems 28
Secure Deployment Checklist 28

11 Installing BRM REST Services Manager

Downloading the BRM REST Services Manager Package 1
Installing BRM REST Services Manager 1
Installing BRM REST Services Manager in GUI Mode 2
Installing BRM REST Services Manager in Silent Mode 7
Creating a Response File 7
Installing BRM REST Services Manager in Silent Mode 7
BRM REST Services Manager Postinstallation Tasks 8
Setting Up a Custom TrustStore 9
Configuring Policies for API Authorization 9
Excluding the BRM REST Services Manager SDK 9
Updating BRM Connection Details 10
Updating the Base URL 10
Modifying BRM REST Services Manager Properties 10
Setting Up Zipkin Tracing in BRM REST Services Manager 11
Connecting BRM REST Services Manager to Oracle Analytics Publisher 11
Configuring BRM REST Services Manager for High Availability 11
Allowing the Use of Deprecated Cyphers 12
Customizing REST Services with the Mapper Framework 12
Introduction to the Mapper File 13
Understanding the Mapper File Structure 14
Examples and Supported Field Attributes 14
Mapper Features 16
Best Practices for Customizing Mapper Files 26
Configuring and Adding Custom Mapper Files 26

Configuring the REST Services Manager Notification Service 28
Overview and Prerequisites for REST Services Manager Notification Service 28
REST Services Manager Notification Service Configuration Process 29
Setting Up Connection Profiles 30
Completing Post-Installation Tasks 36
Performing Verification and Sanity Checks 37
Security Best Practices for REST Services Manager Notification Service 37
Troubleshooting SSL Connection Errors 37

12 Troubleshooting the BRM Installation

Problems Installing the Database 1
Problem: Cannot Start the Oracle Listener 1
Possible Cause 1
Solution 1
Problem: Cannot Connect to the Oracle Database as the System User 1
Possible Causes 1
Solution 2
Problems Installing BRM 2
Problem: An Error Occurred When Selecting an Optional Component to Install 2
Possible Cause 2
Solution 2
Problem: An Error Occurred When pin_setup is Run During Installation 3
Possible Cause 3
Solution 3
Problem: An Error Occurred during the Move Data Process 4
Possible Cause 4
Solution 4
Problem: Errors Are Recorded in the Log File When dm_oracle Starts 4
Possible Cause 4
Solution 4
Problem: An Error Occurs in the dm_fusa.pinlog file during pin_setup of Paymentech Manager 4
Possible Cause 4
Solution 5
Problem: An Error is Recorded in the dm_oracle.pinlog and dm_oracle Fails to Start 5
Possible Cause 5
Solution 5
Problems Connecting to the Database 5
Problem: Pointer to the Wrong Database Name 6
Possible Cause 6
Solution 6

Problem: Cannot Log In to the Database 6
Possible Cause 6
Solution 6
Problem: Out of Memory on Linux 6
Possible Cause 6
Solution 6

13 Installing Multiple BRM Instances on One System for Testing

Installing and Configuring Multiple Instances of BRM on One Machine 1
Creating Multiple Users 1
Installing and Configuring the Oracle Database for Multiple Instances 2
(Optional) Creating Tablespaces for Each User 2
(Optional) Creating Multiple BRM Users 2
Installing Multiple Instances of BRM 3
Configuring Each BRM Instance 4
Reinitializing the Database 4
Reinitializing BRM Server and Optional Component Data 5
Reinitializing Optional Component Data Only 6
Removing BRM Data from the Database 8

14 Uninstalling BRM

Uninstalling the BRM Software 1
Uninstalling Optional Components 1
Uninstalling BRM 2
Uninstalling BRM Client Applications 3
Uninstalling BRM Client Applications on Linux 3
Uninstalling BRM Client Applications on Windows 4
Uninstalling BRM in Silent Mode 4

Part III Upgrading BRM

15 About Upgrading to BRM 15.2

About Upgrading BRM 1
Considerations for Upgrading from BRM 12.0 2
Planning Your Upgrade 3
Identifying Your Upgrade Team 3
Identifying Who Is Affected by the Upgrade 4
Collecting Information about Your System 4
Determining the Impact of New Features 4

Estimating How Long the Upgrade Will Take 5
How Long Will It Take to Run the Database Upgrade Scripts? 5
How Long Will It Take to Plan, Prepare for, Test, and Perform the Upgrade? 5
Updating Your System Environment 5
Creating Test Environments 6
Old Baseline Release 6
Old Customized Release 7
New Baseline Release 7
New Customized Release (Test System) 7
Transferring Customizations to the New Release 7
Upgrading Customized Policy Source Files 8
Updating Configuration Files 8
Updating Database Customizations 8
Modifying the Content of Custom Tables 8
Modifying the Structure of Custom Tables 9
Fixing Standard Database Objects with Nonstandard Object IDs 9
Updating Custom Reports 9
Updating Custom Applications 9
Testing Your Upgraded System 10
Running Old and New Versions in Parallel 10
Testing the BRM Database 10
Troubleshooting Your Upgraded System 10
Testing Checklist 11
Preparing for the Production System Upgrade 12
Upgrading Your Production System 12

16 Upgrading BRM 12.0 to BRM 15.2

About the Upgrade 1
Tasks Involved in the Upgrade Process 1
Preparing for the Upgrade 2
Shutting Down the Current Instance 3
Turning Off BRM Service Authentication and Authorization 3
Creating a Complete Backup of Your BRM 12.0 Data 3
Backing Up BRM Files 3
Backing Up Your BRM and Pipeline Manager 12.0 Databases 4
Creating the Database for Your New Version 5
Upgrading to BRM 15.2 5
Importing Your BRM 12.0 Data 5
Setting the Environment Variables for Upgrade 5
Installing BRM and Pipeline Manager for Upgrade 7
Upgrading Your Data to BRM 15.2 10

Pointing the BRM 15.2 Installation to the Database Schema 10
Upgrading the BRM Database Schema to the BRM 15.2 Schema 10
Upgrading the Schema on Single-Schema Systems 10
Upgrading the Schema on Multischema Systems 12
Upgrading the Pipeline Manager Database Schema to the BRM 15.2 Schema 15
Completing the Upgrade 16
Creating Large Indexes 16
Installing the BRM 15.2 Client Applications 16
Adding Optional Components 17
Updating Configurations 17
Update the Default Settings for the pin_rel Utility 17
Updating the Settings for the pin_virtual_time Utility 17
Upgrading AMM in a Multischema System 18
Restoring Customizations 18
Post-Upgrade Procedures 18
Loading the pin_notify File 19
Copying Files for Compiling CMs 19
Migrating Custom Code to a 64-bit Environment 20
Updating Queue Handling 21
Setting the Environment Variables for Pipeline Manager 21
Configuring SSL for the BRM Database 21
Removing Invalid Objects from the Database 22
Modifying the Root Encryption Key 22
Modifying the Root Encryption Key on Single-Schema Systems 22
Modifying the Root Encryption Key on Multischema Systems 23
Setting the Bill State for Preexisting Bills 25
Migrating Tax Codes 25
Updating pin_inv_doc_gen Infranet.properties 26
Removing the TRIG_CYCLE_DEFERRED_TAX Trigger 26
Preparing to Migrate Legacy Account Data 27

17 Upgrading BRM 15.x to BRM 15.2

About Upgrading to the Later Release 1
Performing a Zero Downtime Upgrade 2
Tasks Involved in the Upgrade Process for a New Release 3
Preparing for the Upgrade to the New Release 4
Shutting Down the Current Instance 4
Backing Up Files 5
Backing Up Your BRM and Pipeline Manager Databases 5
Installing Optional Components 6
Upgrading to the New Release 6

Upgrading BRM for the New Release 7
Upgrading BRM for the New Release in Silent Mode 8
Creating a Response File 8
Upgrading the Release in Silent Mode 9
Upgrading the BRM Database Schema to the New Release 10
Upgrading the Schema on Single-Schema Systems 10
Upgrading the Schema on Multischema Systems 11
Upgrading the Pipeline Manager Database Schema to the New Release 15
Adding Customizations 16
Post-Upgrade Procedures for the New Release 16
Creating Large Indexes 17
Installing Optional Components After the Release Upgrade 17

18 Rolling Back Your BRM Database Upgrade

About Rolling Back Your BRM Database 1
Tasks for the Database Rollback Process 2
About Reusing Your Old BRM Installation 3

19 Troubleshooting the BRM Upgrade

Problem: A Wallet Error Occurred When Upgrading the Database 1
Possible Cause 1
Solution 1
Problem: Missing Table While Upgrading the Database 1
Possible Cause 2
Solution 2

About This Content

This guide provides an overview of the ways to configure an Oracle Communications Billing and Revenue Management (BRM) system. This guide also describes the system requirements and procedures for installing and configuring BRM.

Audience

This guide is intended for system administrators and those involved in planning BRM systems.

Part I

Preparing for BRM Installation

This part provides information about preparing for your Oracle Communications Billing and Revenue Management (BRM) installation. It contains the following chapters:

- BRM Installation Overview
- Planning Your Database Configuration
- System Requirements

1

BRM Installation Overview

Learn about the high-level steps for installing your Oracle Communications Billing and Revenue Management (BRM) system.

Topics in this document:

- About Planning and Installing a BRM System
- Types of BRM Systems
- Installation Overview
- Installing and Configuring a Localized Version of BRM
- Selecting the Database Character Encoding

Before installing BRM, you must understand the following:

- Basic BRM concepts and system architecture. See "BRM System Overview" in *BRM Concepts*.
- Basic database administration concepts. See the Oracle database documentation.

About Planning and Installing a BRM System

To install and use BRM, you must plan your system by performing the following tasks:

- Determine the services you want to provide (for example, email, telephony, and GSM wireless).
- Determine the type of events you want to rate, how you want to rate them (for example, online rating and offline rating), and the payment type for each service (prepaid or postpaid).
- Depending on these decisions, decide on the BRM system to install and configure (typical, complete, or custom), and if you select a custom system, choose the components you want to install.
- Install BRM and the optional components.
- Configure the components to connect to the appropriate servers. For example, configure the Connection Managers (CMs) to point to the Data Managers (DMs), and the DMs to point to the database.

Each BRM component includes a configuration file, such as a **pin.conf**, **.reg**, or **.properties** file, which you edit to set the environment variables and connection parameters for the component. The configuration files include instructions for all the entries. See "Using Configuration Files" in *BRM System Administrator's Guide*.

By default, the BRM installer stores sensitive information, such as database and account passwords, in the Oracle wallet. BRM applications retrieve the passwords from the Oracle wallet.

Types of BRM Systems

The type of BRM system you install depends on your business needs and the licensing agreement you have with Oracle. For more information about BRM system architecture, see "BRM System Architecture" in *BRM Concepts*.

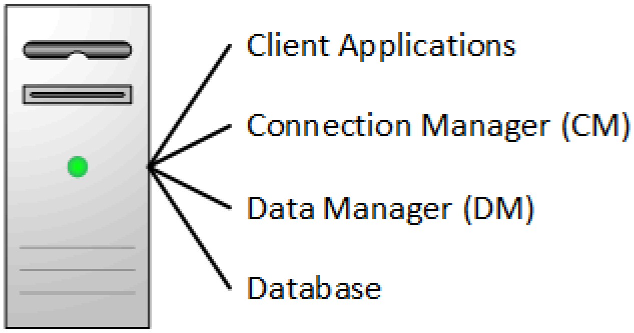
Note

For more information about operating system and software compatibility, see "Overview of Hardware and Software Requirements".

BRM Demonstration or Development System

For a simple demonstration system or a small application development environment, you can run all of the BRM components, including the database server, on a single computer as shown in Figure 1-1.

Figure 1-1 Simple Demonstration System Configuration



For example, you can use a demonstration system for testing your business policies and product offerings. After you create test accounts, you can generate events to ensure that your product offerings are working correctly. For final preproduction testing, however, you should use a test system that mirrors your production system configuration.

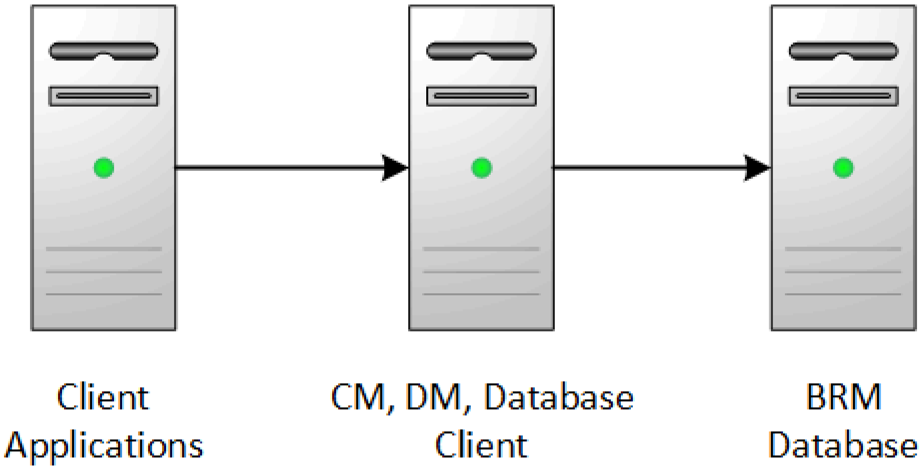
Note

A single-computer configuration is not adequate for a typical production BRM system.

BRM Production System

For best performance on a production system, you must distribute the BRM components among several computers as shown in Figure 1-2.

Figure 1-2   Production System Configuration



By using this basic production configuration, you can expand your BRM system to meet your production needs. You can also add components while BRM is running.

- You can set up multiple machines that each run the CM and DM to increase reliability and performance and to ensure that you continuously have at least one copy of the processes running.
- For additional reliability, you can also set up CM Master Processes (CMMPs) that route requests to an alternate CM when a CM fails. For more information on CMMPs, see "About Connection Manager Master Processes (CMMPs)" in *BRM Concepts*.
- You can use redundant subnets for CM and DM systems, and CMs and DMs can share the same subnet. The only access point is the BRM database, which can be replicated as needed.

The following examples show how you can distribute these components across multiple computers.

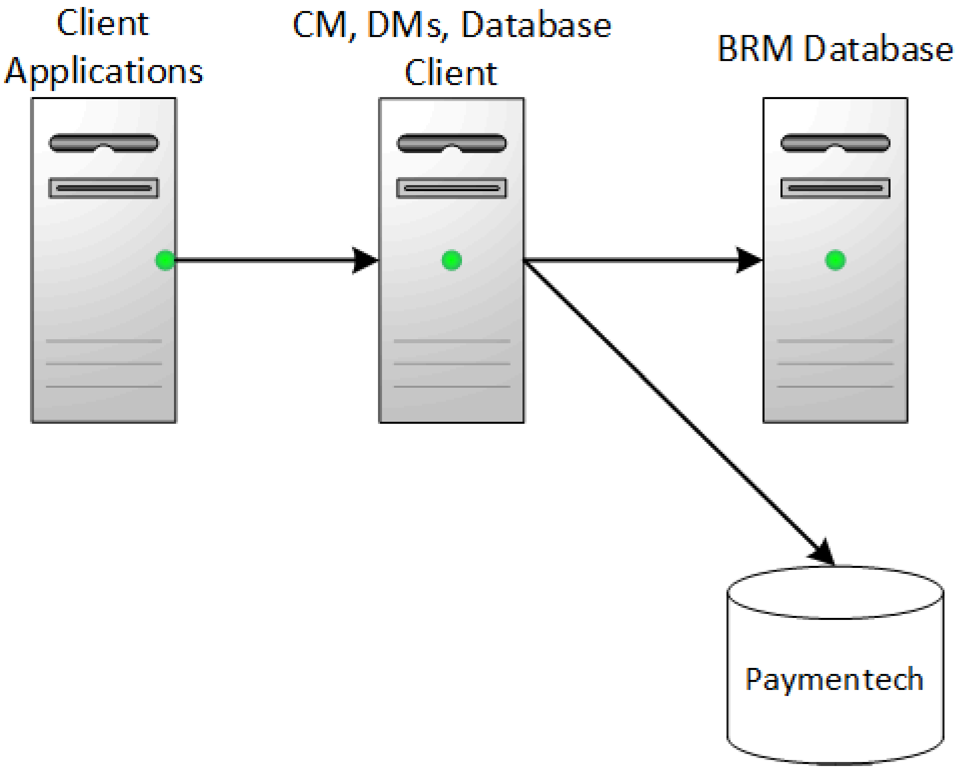
Small Production System

Figure 1-3 shows the minimum level of distributed components. This is a typical configuration for a test deployment.

- All BRM client applications, such as Billing Care or Customer Center, are installed on the same machine.
- The CM and DM are installed on the same machine to optimize performance.
- The database client is installed on the DM machine.

> **Note**
>
> If your database and DM are on separate machines, you must install the database client on the DM machine.

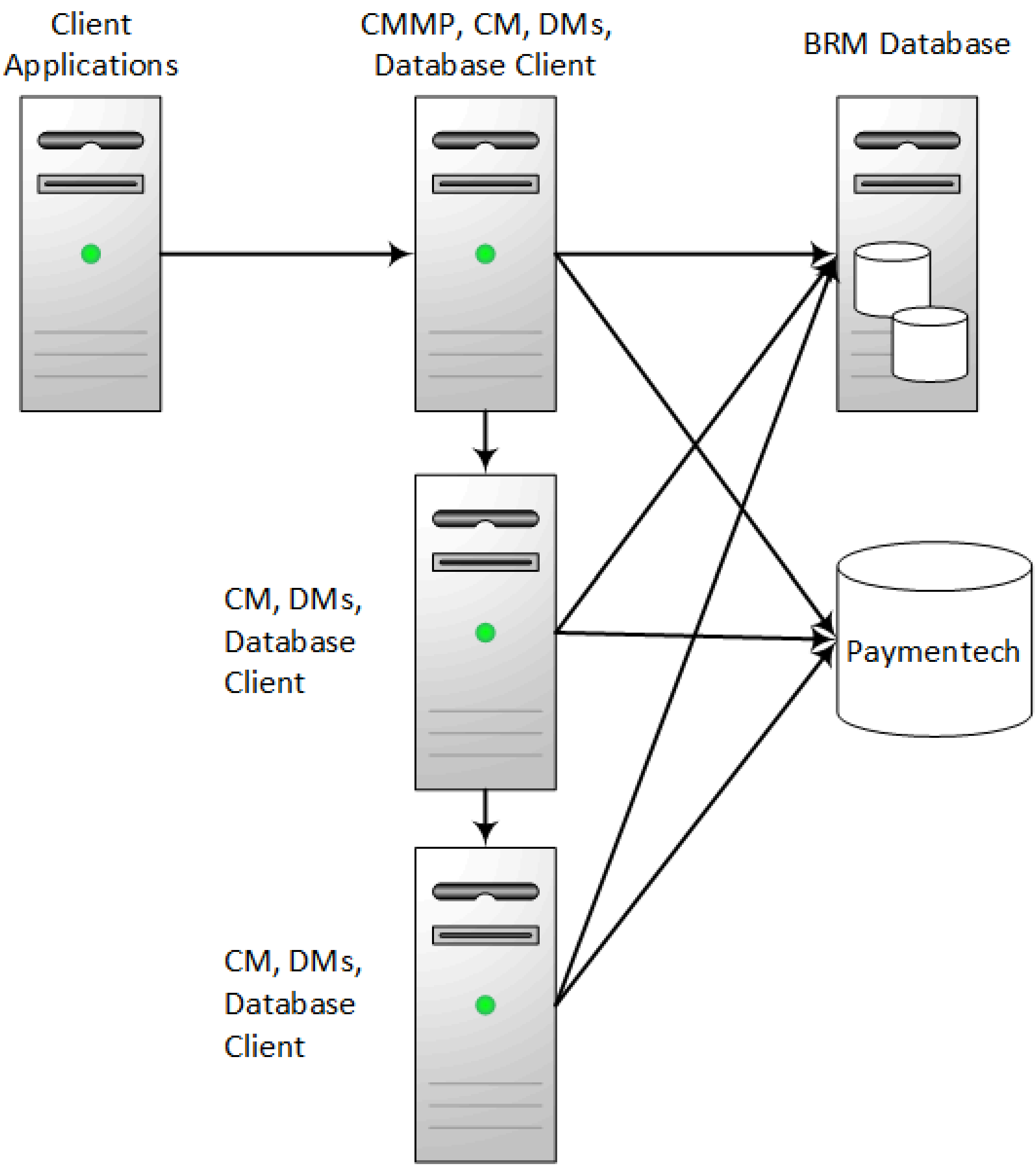
Figure 1-3 Minimum Level of Distributed Components



Mid-Size Production System

Figure 1-4 shows a typical mid-size production system that uses multiple machines for the CM and DM. A CMMP is added to route connections from the client applications to the different CMs.

Figure 1-4   Mid-Size Production System Distributed Components



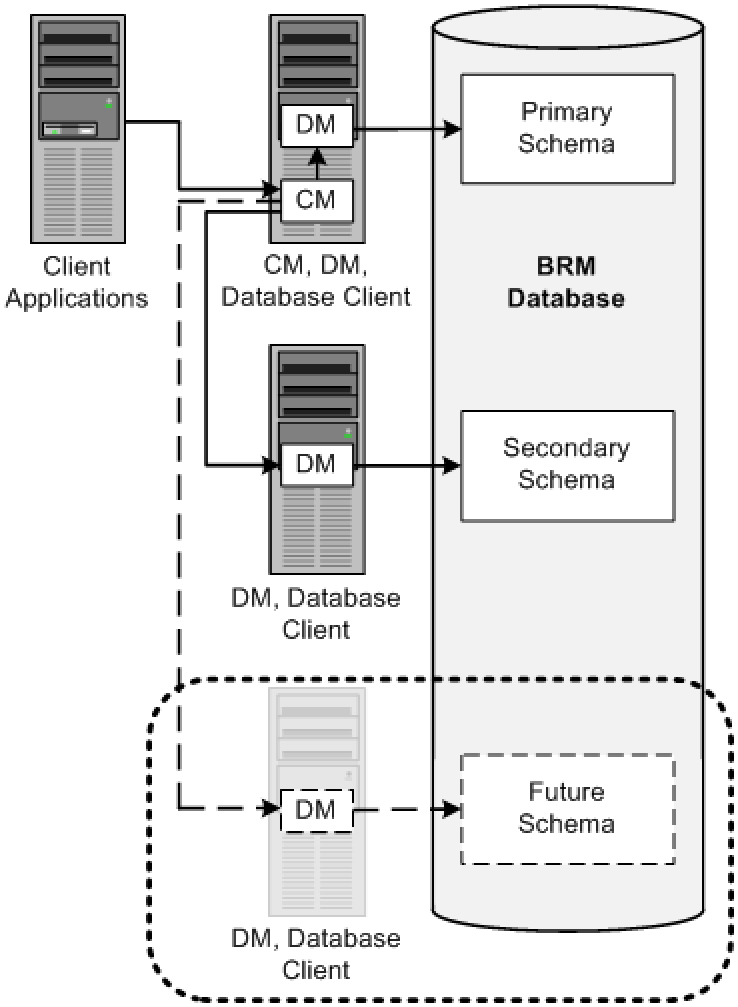
High-Availability Production System

To create a high-availability system, you must install and configure *at least* two instances of each component and then connect each instance to all instances of the appropriate servers. This ensures that if one instance of a component fails, another one is available to process data.

A BRM Multischema Production System

Although BRM works well with a single database schema for BRM data, you can improve scalability and support load balancing by distributing BRM data among multiple schemas as shown in Figure 1-5.

Figure 1-5 BRM Multischema Production System



Designing the Optimal Multischema System

To optimize the benefits of a multischema system, you must decide the following when designing your multischema system:

- How to split BRM applications among multiple schemas
- How to split subscriber data among multiple schemas

To reduce contention at the primary database schema, Oracle recommends the following:

- Set every application to connect to the primary schema.

> **Note**
>
> Any application or utility that directly changes the configuration data, such as PDC and Developer Center, must be connected to the primary schema.

- Distribute accounts among the secondary schemas.

In this configuration, all accounts and associated data are stored in the secondary schemas. Configuration, pricing, audit trail, and uniqueness data are stored and updated in the primary schema and then made available to the secondary schemas.

If the primary schema goes down, you can still access accounts in the secondary schemas. However, to ensure data integrity, you cannot make updates to the primary schema when it is down.

Assigning Accounts to a Database Schema

To set up account distribution for your multischema system, you must first understand that the Multidatabase Manager assigns accounts to a particular database schema based on, in order, account hierarchy, schema status, and schema priority.

Account Hierarchy

The Multidatabase Manager stores data as follows:

- All nonpaying child bill units (**/billinfo** objects) are stored in the same schema as the accounts that own their paying parent bill units.
- All sponsored accounts are stored in the same schema as their sponsor group.
- All accounts associated with a device are stored in the same schema as the device.

This is the first consideration for account storage. Therefore, even if the schema has a status of closed or unavailable, the Multidatabase Manager assigns a child account to the same schema as the parent account.

Database Schema Status

Database schemas are either open, closed, or unavailable. Open schemas are always available for account creation. At installation time, only the primary schema is set to open.

Closed schemas are not used for account creation under most circumstances. Accounts are created in a closed schema only if the account's parent, branded, or sponsoring account belongs to that schema or if all schemas are closed. If all schemas are closed, the Multidatabase Manager chooses a closed schema at random in which to create accounts and continues to create accounts in that schema until a schema becomes open. A schema's status can be manually changed to closed to limit the number of accounts created in the schema, or it can be automatically changed by the Multidatabase Manager when the schema reaches a predefined maximum limit.

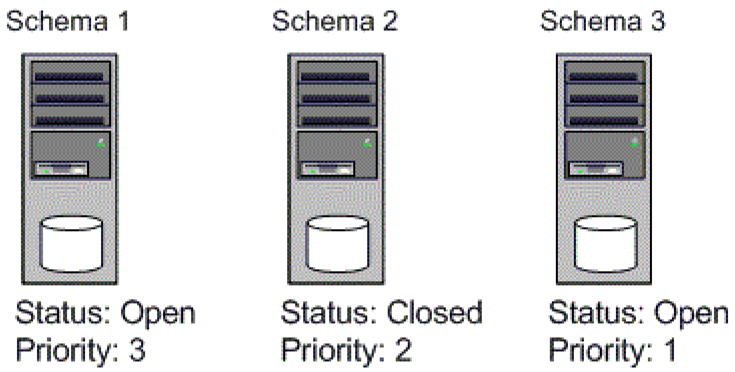
Unavailable schemas are not used for account creation unless the schema contains the account's parent, sponsoring, or branded account. You can change a schema's status to unavailable at any time to suit your system requirements. You might do this, for example, to prevent accounts from being created in the primary schema.

For information on how to set the schema status, see "Setting Database Schema Status" in *BRM System Administrator's Guide*.

Database Schema Priority

Database schema priority determines when customer accounts are created on a particular schema relative to other schemas. The Multidatabase Manager assigns accounts to an open schema with the highest priority number. In the example shown in Figure 1-6, the Multidatabase Manager assigns accounts to schema 3 because it has the highest priority number of all open schemas.

Figure 1-6 Multischema Account Creation Based on Status and Priority



If all schemas have the same priority, the Multidatabase Manager chooses an open schema at random each time it assigns an account. This distributes accounts evenly across all schemas.

For information on how to set the schema priority, see "Setting Database Schema Priorities" in *BRM System Administrator's Guide*.

Installation Overview

The BRM system uses a four-tier architecture. To set up this system, you must ensure that each component of the architecture connects to the other components. This section provides a high-level breakdown of the four-tier installation process.

> **Note**
>
> This section provides guidance only on configuring the components of a BRM system. It does not discuss processes documented for the Oracle database.

To complete your four-tier BRM installation, follow these general steps:

1. Plan your installation. When planning your installation, you do the following:
   - Determine the scale of your implementation, for example, a small test system or a large production system.
   - Determine how many physical machines you need, and which software components to install on each machine.
   - Plan the system topology, for example, how the system components connect to each other over the network.
2. Review system requirements. System requirements include:
   - Hardware requirements, such as disk space
   - System software requirements, such as operating system versions
   - Information requirements, such as IP addresses and host names

   See "Overview of Hardware and Software Requirements".
3. Install and configure your database.

   See "BRM Preinstallation Tasks".

4. Install and configure the BRM software.

   See "Installing BRM".

5. Install the BRM client applications.

   See "Installing BRM Thick Clients".

6. Tune the BRM software for optimal performance.

   See "Improving BRM Performance" in *BRM System Administrator's Guide* and "Planning Your Database Configuration".

Installing and Configuring a Localized Version of BRM

For general information about using a localized version of BRM, see "Using BRM in International Markets" in *BRM Developer's Guide*.

Selecting the Database Character Encoding

For new BRM installations, use the appropriate database character sets when setting up your database.

For Oracle, select AL32UTF8 for the standard character set and for the national character set.

> ⓘ **Note**
>
> BRM supports **AL32UTF8** as its default character set. It also continues to support the **UTF8** character set for backward compatibility. The Unicode character set **AL32UTF8** is recommended for all new BRM deployments.

2

Planning Your Database Configuration

Learn about guidelines for configuring your Oracle Communications Billing and Revenue Management (BRM) database.

Topics in this document:

- Configuring Development or Demonstration System Databases
- Configuring Production System Databases
- Maintaining a BRM Database

Configuring Development or Demonstration System Databases

The following information will assist you in configuring your database for a development or demonstration system.

Tablespace Mapping

In general, you do not need to map tablespaces for development or demonstration systems manually. You can use the default configuration.

For information about managing tablespaces, see "Managing Tablespaces" in *Database Administrator's Guide* in the Oracle database documentation.

Creating Tablespaces

Create the actual tablespaces before running the **pin_setup** configuration script since they require existing tablespaces. You only need to create the two default tablespaces if you have not edited the **pin_tables.values** file to assign tables to nondefault tablespaces.

Configuring init.ora Parameters for Development or Demonstration Systems

Table 2-1 provides guidelines for some database and operating system configuration options specific to BRM. For information on other parameters, see the Oracle database documentation.

For information about memory configuration, see "Tuning Database Memory" in *Database Performance Tuning Guide* and "Managing Memory" in *Database Administrator's Guide* in the Oracle database documentation.

Table 2-1 Configuration Options for Development or Demonstration Systems

Configuration Options	Guidelines
log_buffer	Set the **log_buffer** option to approximately 2 MB. Sample value setting for the **log_buffer** parameter: **2621440**.

Table 2-1 (Cont.) Configuration Options for Development or Demonstration Systems

Configuration Options	Guidelines
open_cursors	A minimum number for **open_cursors** is **1080**. There is no penalty, however, for having this parameter set to a high value. Set **open_cursors** to match the value of the **stmt_cache_entries** entry in the DM Oracle configuration (**pin.conf**) file. If using a statement cache size of **1080**, for example, the **stmt_cache_entries** entry appears as follows in the DM pin.conf file: `- dm stmt_cache_entries -1080`
session_cached_cursors	Set the **session_cached_cursors** option to a nonzero value to enable the database to maintain cursors used in the repeated parsing of SQL statements. Use 150 as a starting point.
db_files	Sample value setting for the **db_files** parameter: **300** to **1020**.
processes	Sample value setting for the **processes** parameter: **320**.
db_block_size	Sample value setting for the **db_block_size** parameter: **8192**.

Configuring Production System Databases

For information about configuring the Oracle database, see the Oracle documentation. In particular, consult the following:

- For information about memory configuration, see "Tuning Database Memory" in *Database Performance Tuning Guide* and "Managing Memory" in *Database Administrator's Guide* in the Oracle database documentation.

  The benchmark testing uses Automatic Shared Memory Management.
- For information about managing tablespaces, see "Managing Tablespaces" in *Database Administrator's Guide* in the Oracle database documentation.
- For information about configuring a high-availability system for best performance, see "Oracle Database High Availability Best Practices" in *High Availability Overview and Best Practices* in the Oracle database documentation.

The following sections contain information about configuring the database for a production system:

- Storage Considerations, Redo, and Undo
- Estimating the Database Size
- International Version Sizing Considerations
- Selecting the Storage Model
- Creating Tablespace Storage
- Creating Tablespaces
- Running Configuration Scripts to Create the BRM Database

Storage Considerations, Redo, and Undo

To determine the storage device type to select, and for other storage considerations, see "I/O Configuration and Design" in *Database Performance Tuning Guide* in the Oracle database documentation.

For information about the redo log buffer, see "Configuring the Redo Log Buffer" in *Database Performance Tuning Guide* in the Oracle database documentation.

Oracle strongly recommends that you run your database in automatic undo management mode instead of using rollback segments. For more information, see "Managing Undo" in *Database Administrator's Guide* in the Oracle database documentation.

Temporary storage should be three to four times the size of the largest table (usually EVENT_T).

Estimating the Database Size

The objects that require the most storage space in a default BRM installation are accounts, bills, and events. The tables that correspond to these objects are also the ones with the most activity:

- ACCOUNT_T
- BAL_GRP_T
- BAL_GRP_BALS_T
- BAL_GRP_SUB_BALS_T
- PURCHASED_DISCOUNT_T
- PURCHASED_PRODUCT_T
- BILL_T
- BILLINFO_T
- ITEM_T
- INVOICE_FORMATS_T
- SERVICE_T

For most customers, the following event tables are very large:

- EVENT_BAL_IMPACTS_T
- EVENT_BILLING_PRODUCT_T
- EVENT_ESSENTIALS_T
- EVENT_T

For Telco service customers, the following tables tend to be very large:

- EVENT_DLAY_SESS_TLCS_T
- EVENT_DLYD_SESSION_TLCO_GPRS_T
- EVENT_DLYD_SESSION_TLCO_GSM_T

Your customizations might create additional large tables. For example, if you store a lot of account profile data, the following tables will also be large:

- PROFILE_T
- PROFILE_SERV_EXTRATING_DA_T
- PROFILE_SERV_EXTRATING_T

Other tables may be large depending on your implementation. Your estimate should not include space gained by archiving.

International Version Sizing Considerations

English databases store VARCHARs as one byte per character. Localized versions of BRM can store these strings in AL32UTF8 or UTF8 format. BRM uses the AL32UTF8 character set. Character strings such as names, addresses, descriptions, and notes that can be manipulated by BRM GUI tools can vary in size. To size the database, you must determine roughly what percentage of the database consists of strings that can vary in size.

Selecting the Storage Model

Choose a storage model based on the total size of your database, which you can determine by summing your data, index, rollback, and temporary tablespaces.

- Use **Test** or **Small** for test or demonstration databases.
- Use **Large** for production databases.

You can set the storage option during installation. See "Installing BRM".

You can set the values used to create the tables using the **pin_tables.values** file. To set these values:

1. Open the *BRM_home***/setup/scripts/pin_tables.values** file in a text editor.
2. To find the block to update, locate the first instance of the following string:

   ```
   PIN_CONF_STORAGE_TEST
   ```

3. Determine the block to update. Table 2-2 contains the mapping between the parameter names and the storage option selected in the installer:

   Table 2-2 Mapping of Table Configurations to Installation Values

pin_tables.values Variables	Storage Model Type
@PIN_CONF_STORAGE_TEST	Test
@PIN_CONF_STORAGE_MINI	Small
@PIN_CONF_STORAGE_MIDSIZE	Medium
@PIN_CONF_STORAGE_HUGE	Large

   For example, if you intend to select a Large configuration in the installer, update the following block:

```
  @PIN_CONF_STORAGE_HUGE = (
        # storage small
        "storage (initial 1m next 1m maxextents unlimited pctincrease 0 )",
        # storage medium
        "initrans 4 storage (initial 1m next 20m maxextents unlimited pctincrease
0 )",
        # storage large
        "initrans 4 storage (initial 1m next 100m maxextents unlimited pctincrease
0 )",
        # storage huge
        "initrans 32 storage (initial 1m next 100m maxextents unlimited pctincrease
0 freelists 40 )",
        # storage small
        "initrans 8 storage (initial 1m next 1m maxextents unlimited pctincrease 0
freelists 8 )",
```

```
        # storage medium
        "initrans 32 storage (initial 1m next 20m maxextents unlimited pctincrease 0
freelists 16 )",
        # storage large
        "initrans 64 storage (initial 1m next 100m maxextents unlimited pctincrease
0 freelists 32 )",
        # storage huge
        "initrans 80 storage (initial 1m next 400m maxextents unlimited pctincrease
0 freelists 40 )"
  );
```

The first four storage clauses are for table storage, and the second four are for index storage.

4. Update the storage clauses with the your required values.

During installation, BRM tables are created using one of the storage models. Therefore, the default storage clause is not used at installation. (The default storage clause specifies the storage parameters to use if no storage clause is provided.) However, if you create custom tables, you must specify a storage clause, or else the default storage clause is used.

Creating Tablespace Storage

For information about managing tablespaces, see "Managing Tablespaces" in *Database Administrator's Guide* in the Oracle database documentation.

In our benchmark implementations, we use locally managed tablespaces with the bigfile option and Oracle Automatic Storage Management (Oracle ASM). This is much easier than managing them manually. If you use the bigfile option, make sure that you monitor the file space usage so that disks don't become full. Following are some sample commands to create data files for a large production environment.

If you are using Oracle ASM, you can find where tablespaces will be created by logging in to SQL*Plus as a user with **sysdba** permissions and running the following command:

```
show parameters DB_CREATE_FILE_DEST;
```

For the first schema, create three bigfile tablespaces:

```
create bigfile tablespace PIN_TABLE DATAFILE SIZE 524288M AUTOEXTEND ON NEXT
8192M MAXSIZE UNLIMITED;
create bigfile tablespace PIN_INDEX DATAFILE SIZE 262144M AUTOEXTEND ON NEXT
8192M MAXSIZE UNLIMITED;
create bigfile temporary tablespace PIN_TEMP TEMPFILE SIZE 65636M  AUTOEXTEND
ON NEXT 4096M MAXSIZE UNLIMITED;
```

Create three additional bigfile tablespaces for each additional schema, changing the tablespace name to be in line with the schema number. For example, commands for the second schema could be:

```
create bigfile tablespace PIN_2TABLE DATAFILE SIZE 524288M AUTOEXTEND ON NEXT
8192M MAXSIZE UNLIMITED;
create bigfile tablespace PIN_2INDEX DATAFILE SIZE 262144M AUTOEXTEND ON NEXT
8192M MAXSIZE UNLIMITED;
create bigfile temporary tablespace PIN_2TEMP TEMPFILE SIZE 65636M
AUTOEXTEND ON NEXT 4096M MAXSIZE UNLIMITED;
```

Adjust the values for **SIZE** and **NEXT** to meet your needs. For a multischema environment, create the same tablespaces for each schema.

For more information about locally managed tablespaces, see "Locally Managed Tablespaces" in *Database Administrator's Guide* in the Oracle database documentation.

For more information about Oracle ASM, see "Introducing Oracle Automatic Storage Management" in *Administrator's Guide* in the Oracle database documentation.

If you are using Oracle RAC, for more information about managing storage, see "Storage Considerations for Oracle Grid Infrastructure and Oracle RAC" in *Grid Infrastructure Installation and Upgrade Guide for Linux* in the Oracle database documentation.

If you use Automatic Storage Management, you generally do not need to map tables to tablespaces.

The size of the database, in turn, determines the minimum number of disks required for the database. Remember to consider disk space required for other purposes, for example, the operating system, BRM, log files, temporary storage, and swap files.

You should have enough disks to avoid performance bottlenecks. In addition, you can increase performance by spreading the most-used tables over multiple disks.

> **Note**
>
> You can add disks and logical devices at any time after BRM has been installed.

After determining the number of disks available for the database, divide the tablespaces among those disks.

Half the remaining space will be used for mirroring. Put mirrors on different disks than their corresponding tables and indexes. The number of mirror logical devices will be equal, in number and size, to their corresponding table and index logical devices.

After allocating disk space for mirrors, divide the remaining disk space using the ratio of 2:1 between tables and indexes. That is, two-thirds of the space will be used for table logical devices and one-third of the space will be used for index logical devices.

Create the logical devices over the remaining physical disks using the above guidelines.

Creating Tablespaces

Create the actual tablespaces before running the **pin_setup** configuration script since they require existing tablespaces. You only need to create the two default tablespaces if you have not edited the **pin_tables.values** file to assign tables to nondefault tablespaces.

Running Configuration Scripts to Create the BRM Database

See "BRM Preinstallation Tasks" for information about creating the BRM database.

Configuring init.ora Parameters

Table 2-3 provides guidelines for some database and operating system configuration options specific to BRM. For information on other parameters, see the Oracle database documentation.

For information about memory configuration, see "Tuning Database Memory" in *Database Performance Tuning Guide* and "Managing Memory" in *Database Administrator's Guide* in the Oracle database documentation.

For information about database performance tuning, see "Oracle Database Configuration Best Practices" in *High Availability Overview and Best Practices* in the Oracle database documentation.

Table 2-3 Configuration Options for Production Systems

Configuration Options	Guidelines
db_writer_processes	You can improve I/O performance by increasing the number of DB writer processes from the default, single process. Setting **db_writer_processes** between 5 and 10 (for the largest systems) can improve I/O throughput. If **db_writer_processes** is set, **dbwr_io_slaves** must not be specified.
DML locks	Use 5000 for DML locks for very heavy workloads. Sample value setting for the **dml_locks** parameter: **5000**.
freelist and **pctfree**	If you are using locally managed tablespaces, you do not need to set these parameters. If you are not using locally managed tablespaces, consider creating tablespaces with additional room for inserting. The storage parameters are pctfree and freelist. Although using freelists requires more disk and memory, insert speed is greatly enhanced. The default is 1. The most active tables should be in tablespaces with at least 10 – 20 freelists, depending on the size of the installation.

open_cursors	A minimum number for **open_cursors** is **1080**. There is no penalty, however, for having this parameter set to a high value.
	Set **open_cursors** to match the value of the **stmt_cache_entries** entry in the DM Oracle configuration (**pin.conf**) file. If using a statement cache size of **1080**, for example, the **stmt_cache_entries** entry appears as follows in the DM pin.conf file:
	`- dm stmt_cache_entries -1080`
	The statement-handle caching performance feature requires a large number of **open_cursors**. Increase the open_cursors parameter to **4192** by adding the following line to the **initSID.ora** file.
	`open_cursors = 4192 (minimum value: statement cache size + number of dm_backends)`

session_cached_cursors	Set the **session_cached_cursors** option to a nonzero value to enable the database to maintain cursors used in the repeated parsing of SQL statements. Use 150 as a starting point.
processes	Sample value setting for the **processes** parameter: **800**.
db_block_size	Sample value setting for the **db_block_size** parameter: **8192**.

Using Optimization

For information about using the optimizer, see "Influencing the Optimizer" and "Optimizer Statistics Concepts" in *SQL Tuning Guide* in the Oracle database documentation.

In our benchmark implementations, we use the **ALL_ROWS** optimizer mode. For performance reasons, it is important to gather good optimizer statistics.

About Using Virtual Columns

Virtual columns are columns whose values are defined by an expression, are computed when you query the data, and are not physically stored in the database. Oracle Database supports virtual columns by default. You can use virtual columns in the BRM database.

> **Note**
>
> BRM's support for virtual columns is deprecated in BRM 15.2 and will be removed in a future release.

Implementations of BRM have shown that a high percentage of the BRM database storage space can be used by the event tables. BRM can use virtual columns in a way that results in space savings for event records. To enable virtual columns in the BRM database, you convert event storable classes (**/event** and its subclasses) in the BRM schema. The savings in database storage applies to event data that the system creates *after* the virtual columns are generated (not to existing event data). Virtual column functionality is transparent to BRM.

For information about virtual columns in general, see the Oracle Database documentation.

For information about generating virtual columns on BRM event tables, see "Generating Virtual Columns on Event Tables" in *BRM System Administrator's Guide*.

3

System Requirements

Learn about the hardware and software requirements for an Oracle Communications Billing and Revenue Management (BRM) system.

Topics in this document:

- Overview of Hardware and Software Requirements
- Assigning Disks for the Operating System and for BRM
- About Network Performance
- About Oracle Exalogic and Oracle Exadata
- Additional Requirements
- Information Requirements for BRM Installation

> **Note**
>
> Before you configure your system, see "Improving BRM Performance" in *BRM System Administrator's Guide* and "Planning Your Database Configuration". These documents contain information about configuring hardware and software for optimal performance with BRM. They also contain a detailed description of memory configuration for the Connection Manager (CM) and Data Manager (DM).

Overview of Hardware and Software Requirements

Running BRM requires the following:

- A compatible operating system.

  See "Operating System Requirements".
- Available disk space. The amount of space required will depend on the type of system (for example, demonstration or production) and the volume and size of transactions.
- Database software.

  See "Database Requirements".
- Network connections.

  See "Network Requirements ".

Operating System Requirements

The BRM software is available for the Linux operating system. To determine which versions of the operating system are currently supported by BRM, see *BRM Compatibility Matrix*.

About Critical Patch Updates

Install all Critical Patch Updates as soon as possible. To download Critical Patch Updates, find out about security alerts, and enable email notifications about Critical Patch Updates, see "Critical Patch Updates, Security Alerts and Bulletins" on the Oracle website.

Database Requirements

BRM supports Oracle databases on the Linux platform.

To run BRM with an Oracle database, you need the following:

- Oracle Enterprise Edition.
- AL32UTF8 database character set.

> ⓘ **Note**
>
> BRM supports **AL32UTF8** as its default character set. You use the Unicode character set **AL32UTF8** for all new BRM deployments.

- (Optional) Oracle Partitioning. You need this to partition the tables in your BRM database. See "Partitioning Tables" in *BRM System Administrator's Guide*.

To determine which versions of the Oracle software are currently supported by BRM, see *Compatibility Matrix*.

Network Requirements

To set up networking and communications channels, you need TCP/IP. You also need a permanent IP address for each computer that hosts a BRM process or application.

Any kind of network connection that supports TCP/IP supports BRM, for example, local area network, virtual private network, and PPP.

TCP/IP

BRM requires TCP/IP on every machine that runs a BRM component, including custom client programs.

> ⓘ **Note**
>
> BRM also needs very large bandwidth to handle traffic between the DM and the database.

IP Addresses

Every computer that runs a BRM component, including the database server, must have its own unique IP address.

> **Note**
>
> BRM uses IP addresses to identify specific machines, so IP addresses cannot be dynamically allocated.

Assigning Disks for the Operating System and for BRM

Allocate separate disks for the BRM software, operating system, and operating system SWAP space. These disks should not be used for any other purpose.

About Network Performance

Any kind of network connection that supports TCP/IP supports BRM (for example, local area network, virtual private network, and PPP).

The network bandwidth needs are relatively simple. BRM has an OLTP footprint (as opposed to Decision Support). This means that transactions are normally small and network traffic will consist of smaller packets. The only requirement for network connectivity is TCP/IP. The real constraint is to have enough bandwidth between the DM systems and the database server; otherwise, the network might become a bottleneck even under moderate loads.

- Use 100BaseT or FDDI between each DM system and the database server. 10 Mbit Ethernets have too little bandwidth to handle heavy DM-to-database-server traffic. To minimize any collisions, the DM-to-database connections should use a separate physical connection (or a switch) and each of these should be a separate network.
- For switch-connected systems, connect all systems in the BRM configuration by using a single switch. For the largest configurations, use gigabit connections.
- When the development systems have multiple network cards, verify that the operating system network routing tables are configured to avoid bottlenecks. By default (depending on the system), output from the database server system may go through one LAN card, even though you have several configured, and even though input comes in through the multiple cards from different DMs. Examine and fix the routing tables as necessary. Ensure that each DM-to-database-server connection is explicit so all traffic between the two machines goes through the single dedicated physical path. The most common environment where you might find this problem is when there are multiple paths between the same sets of systems.
- For best performance, ensure all the systems in the environment are connected to the same hub with no intermediate hops.

About Oracle Exalogic and Oracle Exadata

BRM is supported on Oracle Exalogic and Oracle Exadata systems. You can install BRM server and client software on an Oracle Exalogic system (with 32-bit libraries) on Oracle Linux and the BRM database on an Oracle Exadata system.

You can install and configure the BRM database with Oracle RAC. See the Oracle Exadata documentation for more information about creating an Oracle RAC instance in Oracle Exadata.

When both Oracle Exalogic and Oracle Exadata are available and connected together using InfiniBand network, you can configure a database listener for the network to serve all connection requests from BRM.

Additional Requirements

You need additional software packages to create a complete customer management and billing solution for your business. These packages allow you to take best advantage of BRM functionality:

- Internet software
- Credit card processing software
- Tax calculation software
- Invoice formatting software
- Compilers

You need the following third-party software for installing BRM. See *BRM Compatibility Matrix* for information about the compatible versions.

- Latest certified version of Java Platform, Standard Edition (Java SE), containing Java Development Kit (JDK) and Java Runtime Environment (JRE)

  Important: To protect from security vulnerabilities, ensure you apply the latest critical patch updates. See "About Critical Patch Updates" for more information.

  For information about installing Java, see "Installing Java".
- Latest certified version of Perl

  For information about installing Perl, see "Installing Perl".
- Latest certified version of Apache Kafka

  For instructions on installing Apache Kafka, see "Installing Apache Kafka".

Information Requirements for BRM Installation

This section describes the information that you will be required to provide during the BRM server installation process.

> ⓘ **Note**
>
> Oracle recommends that you record the values in this section for future reference.

Information Requirements for All BRM Installations

This section describes the information requirements common to all BRM server installations.

In each section that follows, you can use the **Value** column of the tables to note the values of the fields for your specific installation.

BRM File Location Details

Table 3-1 lists the BRM file location details you should find out or determine before installing BRM.

Table 3-1 BRM File Location Details

Field	Description	Value
Inventory directory	The full path to the inventory directory, if it is different from the default. The default location of the Oracle inventory is in the /etc/oraInst.loc directory	-
Operating system group name	The name of the operating system group that has write permission to the inventory directory	-
BRM software installation directory	The location you intend to install the BRM software	-
Prerequisite library location	Location of the **ojdbc8.jar** (Oracle Database 19c) or **ojdbc11.jar** (Oracle Database 26ai) file See "Downloading the JDBC Driver" for more information.	-

BRM Oracle Wallet Details

Table 3-2 lists the BRM Oracle wallet details you should find out or determine before installing BRM.

Table 3-2 BRM Oracle Wallet Details

Field	Description	Value
Client wallet password	The client password for the Oracle wallet	-
Root wallet password	The root password for the Oracle wallet	-
Server wallet password	The server password for the Oracle wallet	-

BRM Database SSL Details

Table 3-3 lists the BRM database SSL details you should find out or determine before installing BRM.

Table 3-3 BRM Database SSL Details

Field	Description	Value
Database SSL option	Type of SSL (one-way or two-way) supported by your database, if any	-
TrustStore type	(If your database supports SSL) The type of TrustStore file: SSO or PKCS12	-
TrustStore location	(If your database supports SSL) The directory in which the TrustStore file is located	-
TrustStore password	(If your database supports SSL) The password for the TrustStore	-
KeyStore type	(If your database supports SSL) The type of KeyStore file: SSO or PKCS12	-

Table 3-3 (Cont.) BRM Database SSL Details

Field	Description	Value
KeyStore location	(If your database supports SSL) The directory in which the KeyStore file is located	-
KeyStore password	(If your database supports SSL) The password for the KeyStore	-

BRM Database System User Details

Table 3-4 lists the BRM database system user details you should find out before installing BRM.

Table 3-4 BRM Database System User Details

Field	Description	Value
Host name	The host name or IP address of the machine on which the BRM database is installed	-
Port number	The port number assigned to the BRM database service	-
Database name	The BRM database alias	-
User name	The name of the BRM database system user This user should have the following capabilities on the BRM database: create user, grant any role, grant any privileges, select any table for Enterprise edition, and DBA for XE.	-
Password	The BRM database system user password	-

BRM Database Connection Details

Table 3-5 lists the BRM database connection details you should find out or determine before installing BRM.

Table 3-5 BRM Database Connection Details

Field	Description	Value
User name	The BRM database schema user name	-
Password	The BRM database schema user password	-

BRM Connection Manager Details

Table 3-6 lists the BRM Connection Manager (CM) details you should determine before installing BRM.

Table 3-6 BRM Connection Manager Details

Field	Description	Value
User name	The fully qualified host name or IP address of your CM machine	-
Port number	The port number for your CM	-

Enterprise Application Integration Framework Details

Table 3-7 lists the Enterprise Application Integration (EAI) framework details you should determine before installing BRM.

Table 3-7 Enterprise Application Integration Framework Details

Field	Description	Value
Port number	The port number for your EAI Manager	-

Oracle Data Manager Details

Table 3-8 lists the Oracle Data Manager (DM) details you should determine before installing BRM.

Table 3-8 Oracle Data Manager Details

Field	Description	Value
Database number	The DM database number	-
Port number	The port number for your DM	-

BRM Table and Index Tablespace Details

Table 3-9 lists the BRM table and index tablespace details you should determine before installing BRM.

Table 3-9 BRM Table and Index Tablespace Details

Field	Description	Value
Table tablespace	The name of your data tablespace	-
Index tablespace	The name of your index tablespace	-

Additional Information Requirements for BRM Complete Installation

This section describes the additional information requirements for a complete BRM server installation.

In each section that follows, you can use the **Value** column of the tables to note the values of the fields for your specific installation.

Oracle Connection Manager Proxy Details

Table 3-10 lists the CM proxy details you should determine before installing BRM.

Table 3-10 Oracle Connection Manager Proxy Details

Field	Description	Value
Port number	The port number for your CM Proxy	-

Oracle Connection Manager Master Process Details

Table 3-11 lists the CM Master Process (CMMP) details you should determine before installing BRM.

Table 3-11 Oracle Connection Manager Master Process Details

Field	Description	Value
Port number	The port number for your CMMP	-

Pipeline Schema User Details

Table 3-12 lists the Pipeline schema user details you should determine before installing BRM.

Table 3-12 Pipeline Schema User Details

Field	Description	Value
User name	The name of the Pipeline schema user	-
Password	The Pipeline schema user password	-
Table tablespace	The name of the data tablespace for the pipeline schema	-
Index tablespace	The name of the index tablespace for the pipeline schema	-

BRM Schema User Details

Table 3-13 lists the BRM schema user details you should find out or determine before installing BRM.

Table 3-13 BRM Schema User Details

Field	Description	Value
User name	The name of the BRM database schema user	-
Password	The BRM database schema user password	-
Table tablespace	The name of your BRM data tablespace	-
Index tablespace	The name of your BRM index tablespace	-

Part II

Installing BRM

This part describes how to install Oracle Communications Billing and Revenue Management (BRM) server and client components. It contains the following chapters:

- BRM Preinstallation Tasks
- Installing BRM
- BRM Postinstallation Tasks
- Installing a Multischema System
- Installing BRM Thick Clients
- Installing Pipeline Manager
- Installing Pipeline Configuration Center
- Installing BRM REST Services Manager
- Troubleshooting the BRM Installation
- Installing Multiple BRM Instances on One System for Testing
- Uninstalling BRM

4 BRM Preinstallation Tasks

Learn about the tasks to perform before installing the Oracle Communications Billing and Revenue Management (BRM) system.

To install BRM, you need the following:

- A directory with sufficient space for the BRM software (at least 440 MB).
- A user ID, such as **pin**, with the privileges to access the files under your BRM home directory and to run BRM and third-party applications.
- The BRM-related information that you noted when installing your database.

  See "About Capturing Information When Installing the Oracle Database Software".

You must also perform the following tasks:

- Installing the Database
- Installing Third-Party Tools
- Preparing the Environment

The person installing BRM should be familiar with the following:

- Linux commands and the Linux operating system
- Database installation and configuration
- Network Management Systems
- SQL*Plus
- A Linux text editor, such as **vi** or **nano**

Installing the Database

Learn how to install and configure the database to work with the Oracle Communications Billing and Revenue Management (BRM) system.

Topics in this document:

- Collecting Information Before Installing the Oracle Database Software
- Before Installing Oracle Database Software
- Installing and Configuring Your Oracle Database: Task List
- Installing the Database and Oracle DM on Separate Machines
- Modifying Your Oracle Database Installation

The person installing and configuring the Oracle database should be familiar with the following:

- Linux commands and the Linux operating system
- Database installation and configuration
- Network Management Systems
- SQL*Plus

- A Linux text editor, such as **vi** or **nano**

Collecting Information Before Installing the Oracle Database Software

You will need the following documents while installing and configuring the Oracle database software:

- Oracle documentation:
  - Oracle installation documentation for Linux.
  - The current release notes for your Oracle software.
  - The Oracle Administrator's Guide for instructions about configuring the network.
- A Linux operating system reference guide.

About Capturing Information When Installing the Oracle Database Software

As you install the Oracle software, record the following important information. You will need this information later, when you install BRM.

- Host name of the Oracle database instance.
- IP address of the Oracle database instance.
- Port number of the Oracle database instance.
- User login name and password.
- SID for the BRM database name (the BRM default is **pindb***hostname*).
- Oracle database alias/Global Database Name (for example, **pindb***hostname***.example.com**).

See the tables in "Information Requirements for BRM Installation" for more information about the information that should be collected.

Before Installing Oracle Database Software

Before installing the Oracle database software on your system, verify the following:

- Your system meets the minimum hardware and software requirements.

  See "System Requirements".
- You have all the software required to run BRM with an Oracle database.

  See "Database Requirements".
- You planned your Oracle database according to the guidelines in "Improving BRM Performance" in *BRM System Administrator's Guide* and "Planning Your Database Configuration".

Installing and Configuring Your Oracle Database: Task List

Installing and configuring your Oracle database for BRM includes these major tasks:

> **Note**
>
> This list is not a substitute for the Oracle documentation or the most recent release notes for the database.

1. Installing the Oracle Database Software
2. Creating your BRM Database
3. Setting Environment Variables for the Database
4. Setting Up Your Database for BRM
5. (Optional) Setting Up Oracle RAC

Installing the Oracle Database Software

Install the Oracle database software according to the instructions in the Oracle documentation. When you install the software, pay particular attention to the following requirements.

- Install the **Oracle Enterprise Edition**.
- Choose a **Customized** installation. This option lets you configure Oracle with the AL32UTF8 database character set.

  > **Note**
  >
  > If you installed your database before referring to this document and used the **Complete** installation option, your database might have been installed with another character set. See "Modifying Your Oracle Database Installation" for information on how to modify your existing database.

- To configure discounts in BRM, install the following Oracle components:
  - **Oracle XML DB**. For more information, see the Oracle documentation.
  - **Oracle XML Developer's Kit (XDK)**. For more information, see the Oracle documentation.
- Install Oracle JServer as part of the Oracle Database installation.
- To partition the tables in your BRM database, you must install the **Oracle Partitioning** component. See "Partitioning Tables" in *BRM System Administrator's Guide*.

Creating your BRM Database

Create your database by using the Oracle Database Configuration Assistant, which can be started automatically by the Oracle installer or started manually. If you start it manually, choose the **Custom** option.

> **Note**
>
> BRM supports **AL32UTF8** as its default character set. It also continues to support the **UTF8** character set for backward compatibility. The Unicode character set **AL32UTF8** is recommended for all new BRM deployments.

As you create your database, pay particular attention to the following:

- Specify a **Global Database Name** using the format *DatabaseName***.***DomainName*, where *DatabaseName* is the database name and *DomainName* is the network domain in which the database is located. For example, **pindb***hostname***.example.com**. Most BRM databases use a *DatabaseName* of **pindb***hostname*, but you can use another name.

  > **Note**
  >
  > You can modify your machine's default domain name in the **$ORACLE_HOME/network/admin/sqlnet.ora** file. For information, see your Oracle documentation.

- Specify a **System Identifier (SID)** for your database. For clarity, it should be the same as your Oracle database name. Most BRM databases are named **pindb***hostname*, but you can use another name.
  - Set the **Character Set** to **AL32UTF8**.
  - Set the **National Character Set** to **AL16UTF16**.

For detailed instructions on how to create your database, see the Oracle documentation.

Setting Environment Variables for the Database

Set the environment variables shown in Table 4-1.

Table 4-1 Environment Variables

Environment Variable	Value
LD_LIBRARY_PATH	**$ORACLE_HOME/lib**
NLS_LANG	**American_America.AL32UTF8** You must use **American_America** as the language and territory, regardless of your locale, and the **UTF8** or **AL32UTF8** character set.
TNS_SERVICE_NAME	Set this to the service name of your primary database.

Setting Up Your Database for BRM

For development or demonstration systems, you have the option to configure your database manually or let the BRM installer configure your database automatically. For production systems, you should configure your database manually. Set up your database using one of the following options:

- Using the BRM Installer to Configure Your Database for Demonstration Systems
- Configuring Your Database Manually for Demonstration Systems
- Configuring Your Database Manually for Production Systems

Using the BRM Installer to Configure Your Database for Demonstration Systems

The BRM installer provides the option to automatically configure your database for demonstration or development systems. The installer configures your database by creating one data, one index, and one temporary tablespace; creating the BRM user; and granting connection privileges to the BRM user. If you want the BRM installer to configure your database, your Oracle installation is complete and you can go directly to "Installing BRM".

Configuring Your Database Manually for Demonstration Systems

To configure your database so that it uses additional or larger tablespaces, you can perform the following general tasks:

- Creating BRM Tablespaces
- Creating the BRM user: see "Creating and Configuring pin_user" for details

> **Note**
>
> You can also create the additional tablespaces and the BRM user during the BRM server installation.

Creating BRM Tablespaces

For a simple demonstration BRM system with a single schema, you must create a minimum of three tablespaces for BRM:

- **PIN_TABLE** (for data)
- **PIN_INDEX** (for indexes)
- **PIN_TEMP** (for a temporary tablespace)

To create your tablespaces:

1. If you are not using Oracle Automatic Storage Management (Oracle ASM), create a directory for the tablespaces, such as **/u02/oradata/pindb**.

   This directory is referred to as *db_storge_loc*.

2. If you are using Oracle ASM, you can find where tablespaces will be created by logging in to SQL*Plus as a user with **sysdba** permissions and running the following command:

   ```
   show parameters DB_CREATE_FILE_DEST;
   ```

3. Log in to SQL*Plus as a user with the appropriate permissions and create the data, index, and temporary tablespaces, making sure:

   - Data tablespaces are at least 600 MB with an extent ("next") size of 64 KB.
   - Index tablespaces are at least 400 MB with an extent size of 64 KB.
   - Temporary tablespaces are at least 100 MB with an extent size of 64 KB.

   > **Caution**
   >
   > Do not use any of the following commands or values to create tablespaces for production systems. It will cause performance issues. For information about creating tablespaces for production systems, see "Creating Tablespace Storage"

   Following are sample commands for creating tablespaces if you are using Oracle ASM:

   ```
   create tablespace PIN_TABLE datafile size 600M autoextend on next 64K maxsize
   unlimited;
   ```

```
create tablespace PIN_INDEX datafile size 400M autoextend on next 64K maxsize
unlimited;
create temporary tablespace PIN_TEMP tempfile size 100M autoextend on next 64K
maxsize unlimited;
```

Following are sample commands for creating tablespaces if you are not using Oracle ASM:

```
create tablespace PIN_TABLE datafile 'db_storge_loc/pin_table.dbf' size 600M
autoextend on next 64K maxsize unlimited;
create tablespace PIN_INDEX datafile 'db_storge_loc/pin_index.dbf' size 400M
autoextend on next 64K maxsize unlimited;
create temporary tablespace PIN_TEMP tempfile 'db_storge_loc/pin_temp.dbf' size 100M
autoextend on next 64K maxsize unlimited;
```

Configuring Your Database Manually for Production Systems

To create a production system, you must create multiple tablespaces for the BRM data and indexes. For information on how to estimate your database size, create multiple tablespaces, and map the tablespaces to BRM tables, see "Planning Your Database Configuration".

Setting Up Oracle RAC

To configure Oracle RAC:

1. Set up an Oracle RAC instance for each database schema that you plan to create.

   For more information, see your Oracle RAC documentation.
2. Configure Oracle database services.

   See "Configuring Oracle Database Services".
3. Add entries for the Oracle database services to all **tnsnames.ora** files that will be referenced by the Oracle DMs in your system.

   See "Defining Connections to the Oracle Database Services".

Configuring Oracle Database Services

You use Oracle database services to connect Oracle DMs to Oracle RAC instances. You must map each database service to one Oracle RAC instance.

> **Note**
>
> Oracle RAC systems are typically configured for high availability, so a backup Oracle RAC instance is included in the examples in this section.

For example, for a single-schema system with two RAC instances, configure the database services as shown in Table 4-2.

Table 4-2 Example Database Service Configuration (Single Schema)

Database Service	Oracle RAC Instance	Backup Oracle RAC Instance for High Availability
Service1	Oracle RAC instance 1	Oracle RAC instance 2

To create the services in Table 4-2, log on to any Oracle RAC node as the Oracle database administrator, and run the following command:

```
srvctl add service -d racDatabaseName -s service1 -r racInstanceName1 -a
racInstanceName2 -P Basic
```

For a multischema system, the following example shows the setup for a system with four database schemas and five Oracle RAC instances. In this case, configure the database services as shown in Table 4-3.

Table 4-3 Example Database Service Configuration (Multischema)

Database Service	Oracle RAC Instance	Backup Oracle RAC Instance for High Availability
Service1	Oracle RAC instance 1	Oracle RAC instance 5
Service2	Oracle RAC instance 2	Oracle RAC instance 5
Service3	Oracle RAC instance 3	Oracle RAC instance 5
Service4	Oracle RAC instance 4	Oracle RAC instance 5

To create the services in the preceding table, log on to any Oracle RAC node as the Oracle database administrator, and run the following commands:

```
srvctl add service -d racDatabaseName -s service1 -r racInstanceName1 -a
racInstanceName5 -P Basic
srvctl add service -d racDatabaseName -s service2 -r racInstanceName2 -a
racInstanceName5 -P Basic
srvctl add service -d racDatabaseName -s service3 -r racInstanceName3 -a
racInstanceName5 -P Basic
srvctl add service -d racDatabaseName -s service4 -r racInstanceName4 -a
racInstanceName5 -P Basic
```

The following example shows the commands for a configuration that has four database schemas with a round-robin configuration using Oracle Pluggable Database:

```
srvctl add service -d racDatabaseName -s service1 -r racInstanceName1 -a
racInstanceName2 -P Basic -pdb PDB1
srvctl add service -d racDatabaseName -s service2 -r racInstanceName2 -a
racInstanceName3 -P Basic -pdb PDB2
srvctl add service -d racDatabaseName -s service3 -r racInstanceName3 -a
racInstanceName4 -P Basic -pdb PDB3
srvctl add service -d racDatabaseName -s service4 -r racInstanceName4 -a
racInstanceName1 -P Basic -pdb PDB4
```

For information about the **srvctl** command, see your Oracle RAC documentation.

Defining Connections to the Oracle Database Services

Perform the following procedure in the appropriate **tnsnames.ora** files.

To define connections to the Oracle database services:

1. Open the **tnsnames.ora** file in a text editor.

   By default, that file is in the *Oracle_home***/network/admin/** directory.

2. For each database service, add the following connect descriptor:

   ```
   connectionString =
     (DESCRIPTION=
   ```

```
    (ADDRESS=(PROTOCOL=TCP)(HOST=scanHostname)(PORT=port))
    (CONNECT_DATA=
      (SERVER=DEDICATED)
      (SERVICE_NAME=serviceName)))
```

where:

- *connectionString* is the connection name.

> **Note**
>
> The **sm_database** entry in the Oracle DM **pin.conf** file must match this entry. See "Completing the Installation" for single-schema systems or "Installing BRM on the Primary Installation Machine" and "Installing BRM on a Secondary Installation Machine" for multischema systems.

- *ScanHostname* is the address at which the Single Client Access Name (SCAN) is configured.
- *port* is the port number for the Oracle database. If you do not specify a port number, then the default value of 1521 is used for the TCP port identifier.
- *serviceName* specifies the name of the database service.

> **Note**
>
> The **sm_svcname** entry in the Oracle DM **pin.conf** file must match this entry. It should match the **SERVICE_NAME** entry in the connect descriptor specified in the **connectionString** above. See "Installing BRM on the Primary Installation Machine" and for miultischema systems also see "Installing BRM on a Secondary Installation Machine".

3. Save and close the file.

Installing the Database and Oracle DM on Separate Machines

If you are installing your Oracle database and Oracle DM on separate machines, you must perform the following on the machine containing the Oracle DM:

1. Install the Oracle database client. For the list of supported database clients, see "BRM Software Compatibility" in *BRM Compatibility Matrix*.

> **Note**
>
> For BRM installation to be successful:
>
> - Install the same version of the database server and database client software.
> - Install the 64-bit version of the Oracle database client.

2. Modify the *Oracle_home***/network/admin/tnsnames.ora** file to include entries for connecting to your BRM database.

> **Note**
>
> If you are installing the BRM server and Pipeline Manager on separate machines, ensure that you include the database alias for connecting to the BRM database and the Pipeline Manager database in the *Oracle_home***/network/admin/tnsnames.ora** file. The database alias for each database must be unique.

3. Use SQL*Plus to ensure that you can connect to your database.

For information, see the Oracle database software documentation.

Modifying Your Oracle Database Installation

If you installed your Oracle database before referring to this document and accepted the default options, your database installation might have defaulted to an unsupported character set. If this occurred, you must move your data to a database installation that supports the AL32UTF8 or the UTF8 character set. By default, BRM uses the AL32UTF8 character set.

> **Note**
>
> After you create a database, you *cannot change* the character set.

To export your existing data to a database installation that supports the AL32UTF8 character set:

1. Back up your existing database. See your Oracle documentation for details.
2. Uninstall your existing database. See your Oracle documentation for details.
3. Create a new Oracle database that supports the AL32UTF8 character set.

   See "Creating your BRM Database".
4. Use the Oracle Import utility to import your existing data to the new database.

   See your Oracle documentation for more information.

Installing Third-Party Tools

To install BRM, you need to perform the following tasks:

- Installing Java
- Installing Perl
- Installing Apache Kafka

Installing Java

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

See "BRM Software Compatibility" in *BRM Compatibility Matrix* for the compatible version of Java.

For instructions on installing Java, see Oracle Java documentation.

Installing Perl

Perl is required for the BRM installation. It is not included in the BRM software distribution.

See "BRM Software Compatibility" in *BRM Compatibility Matrix* for the compatible version of Perl.

> **Note**
>
> Ensure you compile Perl on the version of Linux specified in "Linux" in *BRM Compatibility Matrix*.

Before installing Perl (64-bit) on Linux, do the following:

1. Set the environment variables for installing Perl by running the following command:
   - For Bash shell:

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

   - For C shell:

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

To install Perl on Linux:

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

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

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

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

   where *version* is the compatible version of Perl.

3. Run the following command:

   ```
   cd perl-version
   sh Configure-des
           -Accflags=-fPIC -Dcc="gcc -m64 -D_FORTIFY_SOURCE=1"  -Dusethreads -
   Duserelocatableinc
           -Dprefix='perl_path'
   ```

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

> **Note**
>
> If you receive errors, use this command instead, which unsets the LD_LIBRARY_PATH variable:
>
> ```
> sh Configure -des
>         -Dcc="gcc -m64 -D_FILE_OFFSET_BITS=64-D_LARGE_FILE_SOURCE=1" -
> Dusethreads -Dusemorebits
>         -Uuselargefiles -Duserelocatableinc -Dprefix='perl_path'
> ```

4. Run the following make commands:

   ```
   make
   make test
   make install
   ```

5. 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 Apache Kafka

Install the latest compatible version of Apache Kafka, which is required for the BRM installation. It is not included in the BRM software pack.

For the latest compatible version, see "BRM Software Compatibility" in *BRM Compatibility Matrix*.

For instructions on downloading and installing Kafka, see "Apache Kafka Quickstart" on the Apache Kafka website.

Preparing the Environment

You need to perform the following tasks before installing BRM:

- Downloading the JDBC Driver
- (Optional) Creating and Configuring pin_user
- Setting the Environment Variables for Installation
- Setting BRM Wallet Location in sqlnet.ora

Downloading the JDBC Driver

Download the JDBC driver file and save it in a directory on the machine on which you want to install BRM. Note the path to this directory; you are required to specify this path in the Specify Prerequisite Libraries Location window during the BRM installation. The correct file to download is:

- If you are using Oracle Database version 19c, download the latest version of **ojdbc8.jar** for the version 19 database. If you are on a multischema system, also download the latest version of **ojdbc11.jar**. Use the **ojdbc11.jar** file in the **pin_multidb.conf** file here: "Configuring pin_multidb.conf on the Primary Installation Machine" and use **ojdbc8.jar** everywhere else.

- If you are using Oracle Database version 26ai, download only the latest version of **ojdbc11.jar** for the version 26 database.

You can download **ojdbc8.jar** or **ojdbc11.jar** from the following location:

https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html

Creating and Configuring pin_user

You can create *pin_user* before installing BRM or during the BRM installation process. If you create *pin_user* before you install BRM, you must grant execute permission to *pin_user* as well.

Create a new BRM user who can access the Oracle database. The Oracle Data Manager (DM) gains access to the Oracle database by using this Oracle user. This Oracle user owns all the tables created and used by BRM. Usually, the BRM database is set up with the user name **pin**, but you can choose another name.

> **Note**
>
> If you are installing a multischema system, the primary and secondary database schemas must *not* use the same user name. If they do, multischema installation will fail.

To create and configure the BRM user:

1. Log in to your database as user **sysdba**:

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

   where:

   - *password* is the Oracle system database user password.
   - *databaseAlias* is the Oracle system database alias.

2. Create the Oracle user *pin_user* (for example, **pin**), grant the user the resource to connect to Oracle and set the default tablespaces:

   ```
   create user pin_user identified by password;
   grant resource, connect to pin_user;
   alter user pin_user default tablespace PIN_TABLE;
   alter user pin_user temporary tablespace PIN_TEMP;
   ```

3. Grant execute privileges to *pin_user*:

   ```
   grant unlimited tablespaces to pin_user;
   grant alter session to pin_user;
   grant execute on dbms_lock to pin_user;
   grant execute on dbms_aq to pin_user;
   grant execute on dbms_aqadm to pin_user;
   ```

```
grant select on sys.gv_$aq to pin_user;
grant select on v_$session to pin_user;
grant select on sys.dba_2pc_pending to pin_user;
grant create synonym, create any synonym, create public synonym, drop
public synonym, create view, create sequence, create table, create any
index, create procedure to pin_user;
```

> **Note**
>
> If this user is the user for a secondary schema in a multischema installation, you must also run the following command:
>
> ```
> grant select any table to pin_user;
> ```

4. Type **exit** to exit SQL*Plus.

Setting the Environment Variables for Installation

Before installing BRM, you must set the environment variables for BRM and the Java, Perl, and (if used) Apache Kafka software. For information about the latest compatible versions of Java, Perl, and Kafka, see *BRM Compatibility Matrix*.

You must also set the environment variables required by the database. See "Setting Environment Variables for the Database" for details.

To set the environment variables, perform the following for all users you are going to use to install or run the BRM server:

1. Set the PIN_HOME environment variable to the location you intend to install BRM:

   ```
   setenv PIN_HOME install_path
   ```

   where *install_path* is the path to the directory in which you are planning to install BRM.

2. Set the JAVA_HOME environment variable to the directory in which the latest version of Java 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_381**.

3. 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 Java version is not installed in the directory specified by JAVA_HOME.

4. Set the PERL_HOME environment variable:

   ```
   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_36.0/linux**.

5. Set the PATH environment variable:

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

6. Set the HOSTNAME environment variable:

   ```
   setenv HOSTNAME hostName
   ```

   where *hostName* is the fully qualified domain name of the machine on which you want to install the BRM server.

7. If your BRM client wallet is not stored in the default location, set the BRM_CONF_WALLET and BRM_WALLET environment variables to point to the directory in which the BRM client wallet is stored:

   ```
   setenv BRM_CONF_WALLET brm_wallet_path
   setenv BRM_WALLET brm_wallet_path
   ```

8. If you will be using the Kafka Data Manager (DM), set the Kafka environment variables:

   ```
   setenv KAFKA_HOME Kafka_path
   setenv KAFKA_BOOTSTRAP_SERVER_LIST KafkaHost1:port1,KafkaHost2:port2
   ```

   where:

   - *Kafka_path* is the path to the directory in which the Kafka library JARs are installed.
   - *KafkaHost1:port1,KafkaHost2:port2* are the hosts and ports that the Kafka client will connect to in a bootstrap Kafka cluster the first time it starts. You can specify any number of hosts and ports in this list.
     You can alternatively set this list in the **dm_kafka_config.xml** after installing BRM. See "Mapping Business Events to Kafka Topics" in *BRM Developer's Guide*.

Setting BRM Wallet Location in sqlnet.ora

By default, the BRM client wallet is stored in the *BRM_home***/wallet/client** directory. If it is stored in a different location, update the **sqlnet.ora** file accordingly.

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

1. Open the **sqlnet.ora** file on the BRM machine. The file is located in the directory specified by $TNS_ADMIN (typically **$ORACLE_HOME/network/admin**).
2. Add these entries:

   ```
   SQLNET.WALLET_OVERRIDE = TRUE
   WALLET_LOCATION=(
      SOURCE=(METHOD=FILE)
      (METHOD_DATA=(DIRECTORY=brm_wallet_path))
   )
   ```

   where *brm_wallet_path* is the path to the directory in which the BRM client wallet is stored.
3. Save and close the file.

5

Installing BRM

Learn about installing the Oracle Communications Billing and Revenue Management (BRM) software on Linux.

Topics in this document:

- About the BRM Installation Package
- Installing BRM: Task List
- Downloading the BRM Software
- Installing BRM in GUI Mode
- Installing BRM in Silent Mode
- Completing the Installation

This chapter is for network administrators, database administrators, and engineers who install and configure the BRM software. The person installing the software should be familiar with the following topics:

- Linux administration commands and the Linux operating system
- Database configuration
- Network system management

About the BRM Installation Package

The BRM installation package includes the BRM server and BRM client packages. The BRM server package includes all the required and optional BRM server components and applications. You can use the Installer to install one or more of the following individual BRM server components:

- Typical BRM server components. The following are the most commonly used BRM server components:
  - Connection Manager (CM)
  - Oracle Data Manager (DM)
  - Batch Controller
  - Billing and Invoicing utilities
- Connection Manager (CM) Proxy
- Connection Manager (CM) Master Process (CMMP)
- Invoice Data Manager

  Before you install the Invoice Data Manager, make sure you have a separate database schema installed and available to use for storing invoices.
- Kafka Data Manager
- Provisioning Data Manager
- Email Data Manager

- BRM Optional Managers

  You can install BRM with the following optional managers:

  > **Note**
  >
  > You must install the CM and DM before or at the same time as installing the optional managers. Also, make sure that you install the optional managers on the same machine where the CM is installed.

  - Account Migration Manager (AMM)
  - BRM Services Framework Manager
  - BRM SDK

    You can install BRM SDK independent of the BRM server, making it possible to isolate development activities from production servers. For example, you can install BRM SDK on each of the computers used by BRM developers at your site. These developers can share access to test and production BRM servers. However, there are some advantages of installing the BRM SDK on BRM servers. For example, installing the SDK on your BRM server gives you access to sample applications that are not included in the server installation.
  - Conversion Manager
  - Collections Manager

    The BRM Collections Manager server components include opcodes, utilities, and storable classes.
  - Content Manager
  - EAI Manager

    To ensure proper load balancing on your BRM system, you can install EAI Manager on a different machine. However, you must install this on the same machine where the CM is installed.
  - Event Extraction Manager
  - GPRS Manager
  - GSM Manager
  - Inventory Manager
  - IP Address Manager
  - JCA Resource Adapter
  - LDAP Manager
  - MultiDB manager
  - Number Manager
  - Paymentech Manager
  - Partition Upgrade Manager
  - Performance Manager
  - Revenue Assurance Manager
  - Rated Event Loader

  - SIM Manager
  - Roaming Settlement Package
  - Suspense Manager
  - Vertex Manager
  - Voucher Manager
  - Web Services Manager

    Oracle recommends that you install Web Services Manager on the machine on which the typical BRM server components are installed. You must also increase the heap size used by the Java Virtual Machine (JVM) before installing Web Services Manager to avoid "Out of Memory" error messages in the log file.

- BRM Reports
- BRM Reports Optional Managers

  You can install BRM Reports with the following optional managers:

  - MultiDB Reports
  - Collections Reports
  - Content Manager Reports
  - GPRS Reports
  - Number Manager Reports
  - Revenue Assurance Reports
  - Roaming Reports
  - SIM Card Manager Reports
  - SMS Reports
  - Suspense Reports
  - Voucher Reports
- BRM Invoices
- Pipeline Manager
- Pipeline Manager Optional Managers

  You can install Pipeline Manager with the following optional managers:

  - CIBER Roaming Manager
  - Interconnect Manager
  - Pipeline Configuration Manager
  - Pipeline PDK Manager
  - TAP Roaming Manager

For information on the BRM client application package, see "Installing BRM Thick Clients".

Installing BRM: Task List

BRM installation must be performed by a user who has permissions to write to the **oraInventory** directory.

To install the BRM software:

1. Download the BRM package and extract its contents. See "Downloading the BRM Software".
2. Install the BRM software in either GUI mode or silent mode. Installing BRM in silent mode lets you perform a non-interactive installation of BRM. You can use silent mode to install BRM quickly.
   - Installing BRM in GUI Mode
   - Installing BRM in Silent Mode
3. Check your configuration entries and Oracle wallet. See "Completing the Installation".

Downloading the BRM Software

You can download the BRM software from one of the following locations:

- For BRM 15.2.0 or any 15.2.*x* maintenance release: From the Oracle software delivery website (https://edelivery.oracle.com)
- For any 15.2.*x*.*y* patch set release: From the Oracle support website (https://support.oracle.com)

Search for and download the **Oracle Communications Billing and Revenue Management 15.2.*x*.*y*.0** software, where *x* refers to the maintenance release and *y* refers to the patch set release of BRM that you are installing.

The Zip archive includes the installer: **brmserver_15.2.*x*.*y*.0_linux_generic_full.jar**

Installing BRM in GUI Mode

For information about installing BRM in GUI mode, please see the following topics:

- Starting the BRM GUI Installer: All users should read this section.
- Installing a Typical BRM System: Read this section if you want to install only the default BRM components (**Typical** installation type. Recommended for non-production systems.
- Installing All BRM Components: Read this section if you want to install all BRM components (**Complete** installation type).
- Installing Individual BRM Components: Read this section if you want to select the BRM components to install (**Custom** installation type). "Installing CM and DM on Separate Machines" describes a special case of this type of installation.

> **Note**
>
> If you are installing BRM to replace an identical release (for example, to restore a clean version of the package), you must first uninstall the existing installation. See "Uninstalling BRM".

Starting the BRM GUI Installer

To start the BRM GUI installer, go to the directory where you downloaded the BRM software and run the following command:

```
Java_home/bin/java -jar jarFile[ -invPtrLoc invPath/oraInst.loc][ -record -
destinationFile respPath]
```

where:

- *Java_home* is the directory in which you installed the latest compatible Java version.
- *jarFile* is the BRM installer file. For example:

  `brmserver_15.2.0.0.0_linux_generic_full.jar`
- **-invPtrLoc** starts an optional clause to use if you want to use an **oraInventory** directory that already exists in a different location from the one specified in the **/etc/oraInst.loc** file.
- *invPath* is the path to the directory in which the **oraInst.loc** file is located.
- **-record** starts an optional clause to use if you want to create a silent installer response file during the installation.
- *respPath* is the absolute path to the response file.

For example:

```
/tools/Linux/jdk/bin/java -jar brmserver_15.2.0.0.0_linux_generic_full.jar
```

or

```
/tools/Linux/jdk/bin/java -jar brmserver_15.2.0.0.0_linux_generic_full.jar -invPtrLoc /
support/oraInventory/oraInst.loc
```

or

```
/tools/Linux/jdk/bin/java -jar brmserver_15.2.0.0.0_linux_generic_full.jar -record -
destinationFile /brm15/silent/InstComp.txt
```

or

```
/tools/Linux/jdk/bin/java -jar brmserver_15.2.0.0.0_linux_generic_full.jar -invPtrLoc /
support/oraInventory/oraInst.loc -record -destinationFile /brm15/silent/InstComp.txt
```

After the GUI installer starts, the steps for installing BRM depend on the BRM server components you choose to install:

- To install only the default BRM server components, select the **Typical** installation option.

  This option installs the most common BRM server components that are required by default. Use this option for evaluation, demonstration, and functional testing.

  See "Installing a Typical BRM System" for more information.
- To install all BRM server components, select the **Complete** installation option.

  See "Installing All BRM Components" for more information.

  This option installs all the BRM server components. Use this option for development and production systems.
- To choose one or more BRM server components to install each time you run the installer, select the **Custom** installation option.

  This option installs a subset of BRM server components that you select. Recommended for advanced users. Use this option for production systems.

  See "Installing Individual BRM Components" for more information.

  For an overview of the different ways to set up a production system, see "Types of BRM Systems".

Installing a Typical BRM System

The instructions in this section assume that you are installing all of the required BRM server components, including the database, on a single computer.

A Typical BRM system is a self-contained version of BRM that contains only the most typically used BRM components. Use this option for evaluation and demonstration systems.

To install a Typical BRM system, launch the installer and then answer the window prompts:

1. In the Welcome window, click **Next**.
2. If the **oraInst.loc** file is corrupt or not present, the Specify Inventory Directory and Credentials window appears next. Otherwise, go to step 3. In the Specify Inventory Directory and Credentials window, enter the details listed in Table 5-1 and then click **Next**.

Table 5-1 Specify Inventory Directory and Credentials

Field	Description
Inventory Directory	The full path to the inventory directory. This should be an absolute path, not containing environment variables. Do not include the name of the **oraInst.loc** file. The directory and oraInst.loc file will be created if they do not already exist. The default location of the **oraInventory** directory is recorded in the **/etc/oraInst.loc** file.
Operating System Group Name	The name of the operating system group that has write permission to the inventory directory.

3. In the Installation Location window, enter the full path or browse to the directory in which to install BRM. This should be an absolute path, not containing environment variables. Click **Next**.
4. In the Installation Type window, select **Typical**. The lower part of the window displays the components that are included in the Typical installation. Click **Next**.
5. In the Specify Prerequisite Libraries Location window, enter the details listed in Table 5-2 and then click **Next**.

Table 5-2 Specify Prerequisite Libraries Location

Field	Description
Prerequisite Libraries	The absolute path of the **ojdbc8.jar** or **ojdbc11.jar** file. See "Downloading the JDBC Driver" for more information.
Enable SSL	Do one of the following: • To enable secure communication between BRM server components, select the **Enable SSL** check box. • To not use SSL between BRM server components, deselect the check box.

6. In the Specify Wallet Details window, enter the details listed in Table 5-3 and then click **Next**.

Table 5-3 Specify Wallet Details

Field	Description
Client Wallet Password	The password for the client Oracle wallet
Confirm Client Password	The client Oracle wallet password again
Root Wallet Password	The password for the root Oracle wallet
Confirm Root Password	The root Oracle wallet password again
Server Wallet Password	The password for the server Oracle wallet
Confirm Server Password	The server Oracle wallet password again

7. In the Specify BRM Log Location window, enter the full path to the directory in which to store BRM log files and then click **Next**.

8. In the Database SSL Options window, select one of the following:
   - If your database uses one-way SSL, select **Yes One Way**, click **Next**, and go to the next step.
   - If your database uses two-way SSL, select **Yes Two Way**, click **Next**, and go to the next step.
   - If your database does not use SSL, select **No**, click **Next**, and go to step 10.

9. In the Database SSL Information window, enter the details listed in Table 5-4 and then click **Next**.

Table 5-4 Database SSL Information

Field	Description
Truststore Type	The type of TrustStore file for the SSL connection: **SSO** or **PKCS12**
Truststore Location	The directory in which the TrustStore file is located
Truststore Password	The password required to access the certificates from the TrustStore
Keystore Type	The type of KeyStore file that is used for the SSL connection: **SSO** or **PKCS12**
Keystore Location	The directory in which your KeyStore file is located
Keystore Password	The password required to access the certificates from the KeyStore

10. In the Create BRM Database Schema User window, do one of the following and then click **Next**.
    - If you want to create a BRM database schema user during installation, select **Yes**.
    - If you do not want to create the BRM database schema user during installation, select **No** and go to step 12.

11. In the BRM Database System User Details window, enter the details listed in Table 5-5 for connecting to the BRM database and then click **Next**.

Table 5-5 BRM Database System User Details

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

Table 5-5 (Cont.) BRM Database System User Details

Field	Description
Database Name	The BRM database alias, for example, pindb.example.com
User Name	The name of the BRM database system user This user should have the following capabilities on the BRM database: create user, grant any role, grant any privilege, and select any table.
Password	The BRM database system user password

> **Note**
>
> Ensure that the Oracle Database server is in **Running** state. The Installer connects to the Oracle Database server to verify that the information you entered is valid.

12. In the BRM Database Connection Details window, enter the details listed in Table 5-6 and then click **Next**.

Table 5-6 BRM Database Connection Details

Field	Description
Host Name	The host name or IP address of the machine on which the BRM database is installed
Port Number	The port number assigned to the BRM database service
Database Name	The BRM database alias, for example, pindb.example.com
User Name	The BRM database schema user name
Password	The BRM database schema user password
Confirm Password	The BRM database schema user password again This is enabled only if you selected to create a BRM database schema user.

13. In the BRM Root User Details window, enter the details listed in Table 5-7 and then click **Next**. This user has all of the permissions in BRM.

Table 5-7 BRM Root User Details

Field	Description
Root Password	The password for the BRM root user
Confirm Root Password	The BRM root user password again

14. In the BRM User Details window, enter the details listed in Table 5-8 and then click **Next**. This password will be used for all of the default BRM users except the root user. You should update these passwords after the install.

Table 5-8 BRM User Details

Field	Description
User Password	The password for the non-root default BRM users
Confirm user Password	The non-root BRM user password again

15. In the BRM Connection Manager (CM) Details window, enter the details listed in Table 5-9 for connecting to the CM and then click **Next**.

Table 5-9 BRM Connection Manager (CM) Details

Field	Description
Host Name	The host name or IP address of your CM machine
Port Number	The port number for your CM

16. In the Enterprise Application Integration (EAI) Framework Details window, enter the port number for your EAI Manager and then click **Next**.

17. In the Oracle Data Manager (DM) Details window, enter the details listed in Table 5-10 for connecting to the DM database and then click **Next**.

Table 5-10 Oracle Data Manager (DM) Details

Field	Description
Database Number	The DM database number. The default value is **0.0.0.1**. **Note:** In BRM 15.2, you can use only the default **Database Number** for the primary schema.
Port Number	The port number to connect to your DM.
Is Primary DB?	Do one of the following: • Select **Yes** if the DM is being installed in the primary schema and go to step 19. • Select **No** if the DM is not being installed in the primary schema.

18. In the Primary DB details window, enter the details listed in Table 5-11 for the primary database and then click **Next**.

Table 5-11 Oracle Primary DB Details

Field	Description
Database Name	The alias for the primary database
User Name	The user name for the primary database
Password	The password for the primary database user

19. In the Select BRM Data Storage Model window, set the size of your database and then click **Next**.

    - For test or demonstration databases smaller than 700 MB, select **Test**.
    - For demonstration databases smaller than 1.5 GB, select **Small**.
    - For production databases smaller than 30 GB, select **Medium**.
    - For production databases larger than 30 GB, select **Large**.

20. In the BRM Table and Index Tablespace Details window, enter the details listed in Table 5-12 and then click **Next**. If you elected to create the BRM schema user, these tablespaces will be created. Otherwise, the tablespaces should already exist.

Table 5-12 BRM Table and Index Tablespace Details

Field	Description
Table Tablespace	The name of your data tablespace
Index Tablespace	The name of your index tablespace

21. In the Select Drop BRM Database Tables window, do one of the following and then click **Next**.

    - To drop the BRM database tables and reinitialize the database, select **Yes**. Select this option for test systems, for example where an unsuccessful previous installation might have left some tables in the database.

      ⓘ **Note**

      If you select **Yes**, the Installer drops all of the existing BRM database tables on your system. This results in irrecoverable loss of data. Ensure that you have backed up all of your existing data before selecting this option.

      This does not impact the Pipeline Manager database tables.

    - To retain the BRM database tables, select **No**.

      The Installer uses your existing BRM database tables.

22. In the Select BRM Database Partitions window, do one of the following and then click **Next**.

    If you are installing the BRM server for evaluation or demonstration, you can select **No** to disable partitioning of the database and you need not install Oracle Partitioning. For a production system, however, you must install Oracle Partitioning.

    ⚠ **Caution**

    To partition tables, you must have Oracle Partitioning installed. If you select **Yes** but do not have Oracle Partitioning installed, the BRM setup program fails when it tries to create partitions. See "Database Requirements" for more information.

    - To enable partitioning for all event tables in your database, select **Yes**.

      If you select **Yes**, the following classes are automatically enabled for partitioning: event, journal, journal master, newsfeed, and user activity.

      This sets the $ENABLE_PARTITION parameter to **Yes** in the **pin_setup.values** file.

      If you plan to use Rated Event (RE) Loader to load rated events, you must partition your event tables.

    - If you do not want to enable partitioning during installation, select **No** and go to step 24.

You can also enable partitioning after installation. See the discussion about converting nonpartitioned classes to partitioned classes in *BRM System Administrator's Guide*.

23. In the BRM Database Partition Details window, do one of the following and then click **Next**.
    - To add 12 monthly partitions, a historic partition, and a last partition to your event tables, select **Monthly (12 partitions)**.

      > ⓘ **Note**
      >
      > This sets the $SETUP_CREATE_PARTITIONS parameter to **Yes** in the **pin_setup.values** file.

      In the **Non-event tables** field, add the class names for partitioning your non-event tables.
    - To add only a historic partition and a last partition to the event tables, select **Default (2 partitions)**.

      You can use this partitioning layout for a simple test or demonstration system. For a production system, however, you must add purgeable partitions after installation is complete and before the system generates events.
24. In the Select Running the pin_setup Script window, do one of the following and then click **Next**.
    - To run the **pin_setup** script during installation, select **Yes**. Running this script installs the database components.
    - To prevent the **pin_setup** from running during installation, select **No**. Do this if you want to modify the database configuration before installing.

    > ⓘ **Note**
    >
    > If **pin_setup** fails due to invalid database alias or service name, see "Problem: An Error Occurred When pin_setup is Run During Installation ".
25. In the Installation Summary window, review your selections and then click **Install**.

    > ⓘ **Note**
    >
    > After the installation begins, you cannot stop or cancel the installation.

    The Installation Progress window appears, and the installation begins.

    If the BRM database and Pipeline Manager database are installed on different machines, the Pipeline Manager database details will not appear in the installation summary.

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

    For information about BRM installer logs, see "Troubleshooting the BRM Installation".
26. When the installation is done, click **Next**.
27. In the Installation Complete window, click **Finish**.

Installing All BRM Components

For a production system, you optimize BRM performance and availability by installing and running the BRM database on its own computer and the various processes on separate computers.

> **Note**
>
> If you are installing the Oracle DM and your database on separate machines, you must install a database client. For information, see "Installing the Database and Oracle DM on Separate Machines".

To install all BRM components, launch the installer and then answer the window prompts:

1. In the Welcome window, click **Next**.
2. If the **oraInst.loc** file is corrupt or not present, the Specify Inventory Directory and Credentials window appears next. Otherwise, go to step 3. In the Specify Inventory Directory and Credentials window, enter the details listed in Table 5-13 and then click **Next**.

Table 5-13 Specify Inventory Directory and Credentials

Field	Description
Inventory Directory	The full path to the inventory directory. This should be an absolute path, not containing environment variables. Do not include the name of the **oraInst.loc** file. The directory and oraInst.loc file will be created if they do not already exist. The default location of the **oraInventory** directory is recorded in the **/etc/oraInst.loc** file.
Operating System Group Name	The name of the operating system group that has write permission to the inventory directory.

3. In the Installation Location window, enter the full path or browse to the directory in which to install BRM. This should be an absolute path, not containing environment variables. Click **Next**.
4. In the Installation Type window, select **Complete**. The lower part of the window displays the components that are included in the Complete installation. Click **Next**.
5. In the Specify Prerequisite Libraries Location window, enter the details listed in Table 5-14 and then click **Next**.

Table 5-14 Specify Prerequisite Libraries Location

Field	Description
Prerequisite Libraries	The absolute path of the **ojdbc8.jar** or **ojdbc11.jar** file. See "Downloading the JDBC Driver" for more information.
Enable SSL	Do one of the following: • To enable secure communication between BRM server components, select the **Enable SSL** check box. • To not use SSL between BRM server components, deselect the check box.

6. In the Specify Wallet Details window, enter the details listed in Table 5-15 and then click **Next**.

Table 5-15 Specify Wallet Details

Field	Description
Client Wallet Password	The password for the client Oracle wallet
Confirm Client Password	The client Oracle wallet password again
Root Wallet Password	The password for the root Oracle wallet
Confirm Root Password	The root Oracle wallet password again
Server Wallet Password	The password for the server Oracle wallet
Confirm Server Password	The server Oracle wallet password again

7. In the Specify BRM Log Location window, enter the full path to the directory in which to store BRM log files and then click **Next**.

8. In the Database SSL Options window, select one of the following:

   - If your database uses one-way SSL, select **Yes One Way**, click **Next**, and go to the next step.
   - If your database uses two-way SSL, select **Yes Two Way**, click **Next**, and go to the next step.
   - If SSL is disabled in your database, select **No**, click **Next**, and go to step 10.

9. In the Database SSL Information window, enter the details listed in Table 5-16 and then click **Next**.

Table 5-16 Database SSL Information

Field	Description
Truststore Type	The type of TrustStore file for the SSL connection: **SSO** or **PKCS12**
Truststore Location	The directory in which the TrustStore file is located
Truststore Password	The password required to access the certificates from the TrustStore
Keystore Type	The type of KeyStore file that is used for the SSL connection: **SSO** or **PKCS12**
Keystore Location	The directory in which your KeyStore file is located
Keystore Password	The password required to access the certificates from the KeyStore

10. In the Create BRM Database Schema User window, do one of the following and then click **Next**.

    - If you want to create a BRM database schema user during installation, select **Yes**.
    - If you do not want to create the BRM database schema user during installation, select **No** and go to step 12.

11. In the BRM Database System User Details window, enter the details listed in Table 5-17 for connecting to the BRM database and then click **Next**.

Table 5-17 BRM Database System User Details

Field	Description
Host Name	The host name or IP address of the machine on which the BRM database is installed
Port Number	The port number assigned to the BRM database service
Database Name	The BRM database alias
User Name	The name of the BRM database system user This user should have the following capabilities on the BRM database: create user, grant any role, grant any privilege, and select any table.
Password	The BRM database system user password

> ⓘ **Note**
>
> Ensure that the Oracle Database server is in **Running** state. The Installer connects to the Oracle Database server to verify that the information you entered is valid.

12. In the BRM Database Connection Details window, enter the details listed in Table 5-18 and then click **Next**.

Table 5-18 BRM Database Connection Details

Field	Description
Host Name	The host name or IP address of the machine on which the BRM database is installed
Port Number	The port number assigned to the BRM database service
Database Name	The BRM database alias
User Name	The BRM database schema user name
Password	The BRM database schema user password
Confirm Password	The BRM database schema user password again This is enabled only if you selected to create a BRM database schema user.

13. In the BRM Root User Details window, enter the details listed in Table 5-19 and then click **Next**.

Table 5-19 BRM Root User Details

Field	Description
Root Password	The password for the BRM root user
Confirm Root Password	The BRM root user password again

14. In the BRM User Details window, enter the details listed in Table 5-20 and then click **Next**. This password will be used for all of the default BRM users. You should update these passwords after the install.

Table 5-20 BRM User Details

Field	Description
User Password	The password for the non-root default BRM users
Confirm user Password	The non-root BRM user password again

15. In the BRM Connection Manager (CM) Details window, enter the details listed in Table 5-21 for connecting to the CM and then click **Next**.

Table 5-21 BRM Connection Manager (CM) Details

Field	Description
Host Name	The host name or IP address of your CM machine
Port Number	The port number for your CM

16. In the Enterprise Application Integration (EAI) Framework Details window, enter the port number for your EAI Manager and then click **Next**.
17. In the Oracle Connection Manager Proxy Details window, enter the port number for your CM Proxy and then click **Next**.
18. In the Oracle Connection Manager Master Process Details window, enter the port number for your CM Master Process (CMMP) and then click **Next**.
19. In the Oracle Data Manager (DM) Details window, enter the details listed in Table 5-22 for connecting to the DM database and then click **Next**.

Table 5-22 Oracle Data Manager (DM) Details

Field	Description
Database Number	The DM database number. The default value is **0.0.0.1**. **Note:** You can use only the default **Database Number** for the primary schema.
Port Number	The port number to connect to your DM.
Is Primary DB?	Do one of the following: • Select **Yes** if the DM is being installed in the primary schema and go to step 21. • Select **No** if the DM is not being installed in the primary schema.

20. In the Primary DB details window, enter the details listed in Table 5-23 for the primary database and then click **Next**.

Table 5-23 Oracle Primary DB Details

Field	Description
Database Name	The alias for the primary database
User Name	The user name for the primary database
Password	The password for the primary database user

21. In the Select BRM Data Storage Model window, set the size of your database and then click **Next**.

    - For test or demonstration databases smaller than 700 MB, select **Test**.

- For demonstration databases smaller than 1.5 GB, select **Small**.
- For production databases smaller than 30 GB, select **Medium**.
- For production databases larger than 30 GB, select **Large**.

22. In the BRM Table and Index Tablespace Details window, enter the details listed in Table 5-24 and then click **Next**. If you elected to create the BRM schema user, these tablespaces will be created. Otherwise, the tablespaces should already exist.

Table 5-24 BRM Table and Index Tablespace Details

Field	Description
Table Tablespace	The name of your data tablespace
Index Tablespace	The name of your index tablespace

23. In the Select Drop BRM Database Tables window, do one of the following and then click **Next**.

    - To drop the BRM database tables and reinitialize the database, select **Yes**. Select this option for test systems, for example where an unsuccessful previous installation might have left some tables in the database.

      > **Note**
      >
      > If you select **Yes**, the Installer drops all of the existing BRM database tables on your system. This results in irrecoverable loss of data. Ensure that you have backed up all of your existing data before selecting this option.
      >
      > This does not impact the Pipeline Manager database tables.

    - To retain the BRM database tables, select **No**.

      The Installer uses your existing BRM database tables.

24. In the Select BRM Database Partitions window, do one of the following and then click **Next**.

    If you are installing the BRM server for evaluation or demonstration, you can select **No** to disable partitioning of the database and you need not install Oracle Partitioning. For a production system, however, you must install Oracle Partitioning.

    > **Caution**
    >
    > To partition tables, you must have Oracle Partitioning installed. If you select **Yes** but do not have Oracle Partitioning installed, the BRM setup program fails when it tries to create partitions. See "Database Requirements" for more information.

    - To enable partitioning for all event tables in your database, select **Yes**.

      If you select **Yes**, the following classes are automatically enabled for partitioning: event, journal, journal master, newsfeed, and user activity.

      This sets the $ENABLE_PARTITION parameter to **Yes** in the **pin_setup.values** file.

      If you plan to use Rated Event (RE) Loader to load rated events, you must partition your event tables.

- If you do not want to enable partitioning during installation, select **No** and go to step 26.

  You can also enable partitioning after installation. See the discussion about converting nonpartitioned classes to partitioned classes in *BRM System Administrator's Guide*.

25. In the BRM Database Partition Details window, do one of the following and then click **Next**.

    - To add 12 monthly partitions, a historic partition, and a last partition to your event tables, select **Monthly (12 partitions)**.

      > **Note**
      >
      > This sets the $SETUP_CREATE_PARTITIONS parameter to **Yes** in the **pin_setup.values** file.

      In the **Non-event tables** field, add the class names for partitioning your non-event tables.

    - To add only a historic partition and a last partition to the event tables, select **Default (2 partitions)**.

      You can use this partitioning layout for a simple test or demonstration system. For a production system, however, you must add purgeable partitions after installation is complete and before the system generates events.

      In the **Non-Event Tables** field, add the class names for partitioning your non-event tables and then click **Next**.

26. In the Oracle Data Manager (DM) Email window, enter the port number for the email DM and then click **Next**.

27. In the Oracle Data Manager (DM) Invoice window, enter the port number for the invoice DM and then click **Next**.

28. In the Sample Pricing Data window, do one of the following and then click **Next**.

    - If you want to load the sample pricing data during installation, select **Yes**.
    - If you do not want to load the sample pricing data during installation, select **No**.

29. In the Select Running the pin_setup Script window, do one of the following and click **Next**.

    - To run the **pin_setup** script during installation, select **Yes**. Running this script installs the database components.
    - To prevent the **pin_setup** from running during installation, select **No**. Do this if you want to modify the database configuration before installing.

    > **Note**
    >
    > If **pin_setup** fails due to invalid database alias or service name, see "Problem: An Error Occurred When pin_setup is Run During Installation ".

30. In the Create Pipeline Schema User window, do one of the following and then click **Next**.

    - If you want to create a pipeline schema user during installation, select **Yes**. Skip step 34 when you get to it.
    - If you already have a pipeline schema user, select **No**. Skip step 33 when you get to it.

31. In the Pipeline Database System User Details window, enter the details listed in Table 5-25 for connecting to the pipeline database and then click **Next**.

Table 5-25 Pipeline Database System User Details

Field	Description
Host Name	The host name or IP address of the machine on which the Pipeline Manager database is installed
Port Number	The port number assigned to the Pipeline Manager database service
Database Name	The Pipeline Manager database alias
User Name	The name of the Pipeline Manager database system user This user should have the following capabilities on the Pipeline Manager database: create user, grant any role, grant any privileges, select any table.
Password	The Pipeline Manager database system user password

> **Note**
>
> Ensure that the Oracle Database server is in **Running** state. The Installer connects to the Oracle Database server to verify that the information you entered is valid.

32. In the BRM Database System User Details window, enter the details listed in Table 5-26 for connecting to the BRM database and then click **Next**.

Table 5-26 BRM Database System User Details

Field	Description
Host Name	The host name or IP address of the machine on which the BRM database is installed This field may be prepopulated with information you provided earlier in the installation.
Port Number	The port number assigned to the BRM database service This field may be prepopulated with information you provided earlier in the installation.
Database Name	The BRM database alias This field may be prepopulated with information you provided earlier in the installation.
User Name	The name of the BRM database system user This user should have the following capabilities on the BRM database: create user, grant any role, grant any privileges, select any table. This field may be prepopulated with information you provided earlier in the installation.
Password	The BRM database system user password

> **Note**
>
> Ensure that the Oracle Database server is in **Running** state. The Installer connects to the Oracle Database server to verify that the information you entered is valid.

33. In the Pipeline Schema User Details window, enter the details listed in Table 5-27 to create a pipeline schema user and click **Next**:

Table 5-27 Pipeline Schema User Details

Field	Description
Host Name	The host name or IP address of the machine on which the Pipeline Manager database is installed
Port Number	The port number assigned to the Pipeline Manager database service
Database Name	The Pipeline Manager database alias
User Name	A pipeline schema user name
Password	The password for the pipeline schema user
Confirm Password	The password for the pipeline schema user again
Table Tablespace	The name of the data tablespace for the pipeline schema
Index Tablespace	The name of the index tablespace for the pipeline schema

34. In the Pipeline Schema User Details window, enter the enter the details listed in Table 5-28 to connect to the existing pipeline schema and click **Next**:

Table 5-28 Pipeline Schema User Details

Field	Description
Host Name	The host name or IP address of the machine on which the Pipeline Manager database is installed
Port Number	The port number assigned to the Pipeline Manager database service
Database Name	The Pipeline Manager database alias
User Name	The pipeline schema user name
Password	The password for the pipeline schema user
Table Tablespace	The name of the data tablespace for the pipeline schema
Index Tablespace	The name of the index tablespace for the pipeline schema

35. In the BRM Schema User Details window, enter the enter the details listed in Table 5-29 to connect to the existing BRM database schema and click **Next**:

Table 5-29 BRM Schema User Details

Field	Description
User Name	The BRM schema user name This field may be prepopulated with information you provided earlier in the installation.
Password	The BRM for the pipeline schema user

Table 5-29 (Cont.) BRM Schema User Details

Field	Description
Table Tablespace	The name of the data tablespace for the BRM schema This may be prepopulated with information you provided earlier in the installation.
Index Tablespace	The name of the index tablespace for the BRM schema This may be prepopulated with information you provided earlier in the installation.

36. In the Installation Summary window, review your selections and then click **Install**.

    > **Note**
    >
    > After the installation begins, you cannot stop or cancel the installation.

    The Installation Progress window appears, and the installation begins.

    If the BRM database and Pipeline Manager database are installed on different machines, the Pipeline Manager database details will not appear in the installation summary.

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

    For information about BRM installer logs, see "Troubleshooting the BRM Installation".

37. When the installation is done, click **Next**.
38. In the Installation Complete window, click **Finish**.

Installing Individual BRM Components

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

You can install a subset of BRM components, such as the CM and DM, on a single machine to save disk space.

> **Note**
>
> - If you are installing the CM and DM on separate machines, see "Installing CM and DM on Separate Machines" for instructions.
> - If you are installing additional BRM instances with Pipeline Manager for configuring high availability, ensure that you also select the **Upgrade Manager Framework** in the Available Product Components window. This ensures that a separate Pipeline Manager schema is not created and allows you to use the Pipeline Manager schema of the primary instance.
>
>   After installing the additional BRM instance, you can remove the upgrade files in the *BRM_home* directory.

To install individual BRM components, launch the installer and then answer the window prompts:

1. In the Welcome window, click **Next**.
2. If the **oraInst.loc** file is corrupt or not present, the Specify Inventory Directory and Credentials window appears next. Otherwise, go to step 3. In the Specify Inventory Directory and Credentials window, enter the details listed in Table 5-30 and then click **Next**.

Table 5-30 Specify Inventory Directory and Credentials

Field	Description
Inventory Directory	The full path to the inventory directory. This should be an absolute path, not containing environment variables. Do not include the name of the **oraInst.loc** file. The directory and oraInst.loc file will be created if they do not already exist. The default location of the **oraInventory** directory is recorded in the **/etc/oraInst.loc** file.
Operating System Group Name	The name of the operating system group that has write permission to the inventory directory.

3. In the Installation Location window, enter the full path or browse to the directory in which to install BRM. This should be an absolute path, not containing environment variables. Click **Next**.
4. In the Installation Type window, select **Custom**. The lower part of the window displays the components that are included in the Complete installation, and you can choose which of them to install in the next window. Click **Next**.
5. In the Available Product Components window, select the components to install, and deselect any other components that you do not want to install. Click **Next**.

   > **Note**
   >
   > You cannot deselect a component if it is required to install any of the selected components.

   If you are installing an optional component for the first time after an upgrade, and you see an error similar to the following, see "Problem: An Error Occurred When Selecting an Optional Component to Install".

```
The distribution Billing Revenue Management 15.2.0.0.0 contains incompatible
features with the following:
Billing Revenue Management 15.0.1.0.0 [CORE 15.2.0.0.0->CORE 15.0.1.0.0 (CORE
15.2.0.0.0->[CORE 15.0.1.0.0])]
```

6. If a window other than the Summary window appears, provide the requested information and then click **Next**.

   For the description of the fields displayed, see "Installing All BRM Components". Continue moving through the windows until the Summary window appears.

   Your responses are written to the *BRM_home*/**setup/pin_setup.values** file.

7. When the Installation Summary window appears, review your selections and then click **Install**.

8. The Installation Progress window appears. When installation completes, click **Next**.

   > **Note**
   >
   > After the installation begins, you cannot stop or cancel the installation.

   If the BRM database and Pipeline Manager database are installed on different machines, the Pipeline Manager database details will not appear in the installation summary.

9. The Installation Complete window 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.

   For information about BRM installer logs, see "Troubleshooting the BRM Installation".

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

Installing CM and DM on Separate Machines

To install CM and DM on separate machines:

1. Install DM. For instructions, see "Installing Individual BRM Components".
2. If you enabled SSL for the DM, do the following:
   a. Open the DM configuration file (*BRM_home*/**sys/dm/pin.conf**) file in a text editor.

   b. Search for **enable_ssl** entry:

   ```
   - dm enable_ssl 1
   ```

   c. Set the **enable_ssl** entry to **0**:

   ```
   - dm enable_ssl 0
   ```

   d. Save and close the file.
3. Start all the DMs installed. See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.
4. Install CM with SSL enabled. For instructions, see "Installing Individual BRM Components".
5. On the machine on which you installed the DM in step 1, do the following:
   a. Open the DM configuration file (*BRM_home*/**sys/dm/pin.conf**) file in a text editor.

   b. Search for **enable_ssl** entry:

   ```
   - dm enable_ssl 0
   ```

   c. Set the **enable_ssl** entry to **1**:

```
- dm enable_ssl 1
```

   d. Save and close the file.

6. Start the DM processes.
7. On the machine on which you installed the CM in step 4, start the CM processes.

For instructions on starting the DM and CM processes, see "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

Installing BRM in Silent Mode

The silent installation uses a response file in which you have set installation information. To obtain the response file, you run the GUI installer for the first install. The GUI installer generates a response file that contains the key-value pairs based on the values that you specify during the GUI installation. You can then copy and edit the response file to create additional response files for installing the BRM server on different machines.

> **Note**
>
> If you are installing BRM to replace an identical release (for example, to restore a clean version of the package), you must first uninstall the existing installation. See "Uninstalling BRM".

Creating a Response File

To create a response file:

1. Create the response file by doing one of the following:
   - Create a copy of the response file that was generated during the GUI installation. See "Installing a Typical BRM System" for more information.

     > **Note**
     >
     > The GUI Installer does not store passwords provided during installation in the response file. You must manually add the passwords after creating a copy of the response file.

   - Create a response file using the template by running the following command:

     ```
     Java_home/bin/java -jar jarFile -getResponseFileTemplates
     ```

     where:

     - *Java_home* is the directory in which you installed the latest compatible Java version.
     - *jarFile* is the BRM installer file. For example:

       ```
       brmserver_15.2.0.0.0_linux_generic_full.jar
       ```

     A response file is created with the default values.

You can create as many response files as needed.

2. Open the file in a text editor.
3. Modify the response file you copied by specifying the key-value information for the parameters you want in your installation.

> **Note**
>
> - The response file template contains guidelines and examples on how to enter the values in the parameters.
> - The Installer treats incorrect context, format, and type values in a response file as if no value were specified.

4. Save and close the response file.

Performing a Silent Installation

To perform a silent installation:

1. Create a response file. See "Creating a Response File".
2. Copy the response file you created to the machine on which you are running the silent installation.
3. On the machine on which you are running the silent installation, go to the directory where you downloaded the BRM software, and run the following command:

   ```
   Java_home/bin/java -jar jarFile -debug -invPtrLoc Inventory_home/oraInventory/
   oraInst.loc [parameter=value] -responseFile path -silent
   ```

   where:

   - *Java_home* is the location of the compatible version of Java.
   - *jarFile* is the BRM installer file. For example:

     `brmserver_15.2.0.0.0_linux_generic_full.jar`

   - *Inventory_home* is the location of the Oracle inventory.
   - *parameter* is the name of an installation parameter.
   - *value* is the value of the installation parameter.
   - *path* is the absolute path to the response file.

   For example:

   ```
   $JAVA_HOME/bin/java -jar brmserver_15.2.0.0.0_linux_generic_full.jar -debug -
   invPtrLoc Inventory_home/oraInventory/oraInst.loc INSTALL_TYPE=Complete -
   responseFile /tmp/brm_complete.rsp  -silent
   ```

   The installation runs silently in the background.

> **Note**
>
> For a complete list of the available parameters for the [*parameter*=*value*] clause, and for information about optional debugging parameters, run the following command:
>
> ```
> Java_home/bin/java -jar jarFile -help
> ```

The Installer checks for all required software and writes errors to a log file if it detects any missing or unavailable components or if any connectivity issues occur.

For information about BRM installer logs, see "Troubleshooting the BRM Installation".

Completing the Installation

After BRM has been installed on each machine, check the configuration entries and the BRM root key stored in the client wallet. You can verify the entries by doing the following:

1. Go to the directory in which you installed the BRM server, and source the **source.me** file:

   Bash shell:

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

   C shell:

   ```
   source source.me.csh
   ```

2. Run the following command:

   ```
   orapki wallet display -wallet client
   ```

   The entries stored in the client wallet appear.

   If the entries do not appear, you must store the entries manually in the client wallet by using the **pin_crypt_app** utility.

3. Verify the following entries:

   - The **dm_pointer** entry is stored for each DM in your system and they contain the DM machine's host name or IP address. Any additional **dm_pointer** entries include the correct host name.
   - The DM (**sm_database**) points to the correct database.

Installation is now complete. If you encountered installation problems, verify that the settings in the **pin_setup.values** file on each computer point to the correct CM and DM.

6

BRM Postinstallation Tasks

Learn how to perform postinstallation tasks, such as changing your database partitions and running the **pin_setup** script, after installing the Oracle Communications Billing and Revenue Management (BRM) software.

Topics in this document:

- Scheduling Backups
- Referencing Environment Variables in pin.conf Files
- Postinstallation Tasks for a BRM Non-Production System
- Postinstallation Tasks for a BRM Production Installation
- What's Next?

Scheduling Backups

You should back up the database every night. You can choose a backup solution from your database manufacturer or from a third party. To keep your database always running, ready to respond to online events, use the database online backup utilities for routine backups.

You should also back up your system files (programs, scripts, source code, and documentation) and your data files and keep the backups in a secure, off-site location. To be safe, you should keep at least three iterations of your system backups and at least one month's worth of daily backups. Storage media are usually less expensive than customer problems.

Verify the data and system backup files to ensure that you can recover the data. At times, this secondary system can also serve as a fully functional test system. If BRM releases a patch that can significantly affect your installation, it is important to try it on a test system before installing it in your production system.

> ⓘ **Note**
>
> The only way to verify the data and system backup files is to restore them to another location to ensure that there are no errors in reading, writing, or formatting.

Referencing Environment Variables in pin.conf Files

To prepare for migrating **pin.conf** files to other systems at a future time, you can reference certain environment variables from within the **pin.conf** files, as in this example:

```
- cm fm_module ${PIN_HOME}/lib/fm_utils/$
{LIBRARYEXTENSION} fm_utils_config fm_utils_init pin
```

For more information, see "Preparing for Platform Migration by Using Variables in pin.conf Files" in *BRM System Administrator's Guide*.

Postinstallation Tasks for a BRM Non-Production System

After installing the BRM server, if you have not performed the following tasks during installation, do the following and then run the **pin_setup** script. See "Running the pin_setup Script" for more information.

- (Optional) To further configure BRM, you can edit the *BRM_home***/setup/pin_setup.values** file. See "Editing the pin_setup.values File" for more information.

Do the following *after* running the **pin_setup** script:

- (Optional) Change your database partitions as required. See "Changing Your Database Partitions" for more information.
- If you installed BRM server and Pipeline Manager on separate machines or directories, set the BRM_WALLET and PIN_HOME variables in the BRM server to point to the client wallet and **JARs** in the *Pipeline_home* directory, where *Pipeline_home* is the directory in which Pipeline Manager is installed. See "Setting Environment Variables for Pipeline Manager" for more information.
- If you installed BRM server and Pipeline Manager on separate machines, update the configuration entries for Pipeline Manager in the client wallet manually. See "Updating Client Wallet for Pipeline Manager" for more information.
- Copy the makefiles required for compiling the Connection Manager and Facilities Module instances. See "Copying Files for Compiling CM and FM Modules" for more information.
- Edit and load the event notification list. See "Editing and Loading the Event Notification List" for more information.
- Load the tailor-made stored procedure to create a tailor-made plan. See "Loading the Tailor-Made Stored Procedure" for more information.

Editing the pin_setup.values File

To further configure BRM, such as by changing the default currency and country, you edit the *BRM_home***/setup/pin_setup.values** file. This file stores the information you provided to the installer and a number of database and add-on component parameters.

The entries you need to edit in the **pin_setup.values** file depend on which components you install on the current machine. For example, if you are installing the Email DM on the current machine, edit the **$DM_EMAIL** entries.

The following examples show the important entries to check in three configurations:

- The Connection Manager (CM) and Data Manager (DM) reside on separate machines.
- The system contains multiple Oracle DMs connected to one database.
- Invoices are stored in a separate database schema.

Example 6-1 The CM and DM Reside on Separate Machines

Machines that contain a CM must include the correct port number and host name of each DM in the system. This is especially critical when DMs reside on separate machines from the CM.

For example, you might install the Oracle DM and Email DM on a separate machine from the CM. In this case, you must modify the parameters in the CM machine's **pin_setup.values** file as shown in Table 6-1.

Table 6-1 CM Machine's pin_setup Values

Entry	Description
$DM_ORACLE{'port'}	Must contain the port number of the Oracle DM.
$DM_ORACLE{'hostname'}	Must contain the host name of the machine running the Oracle DM.
$DM_EMAIL{'port'}	Must contain the port number of the Email DM.
$DM_EMAIL{'hostname'}	Must contain the host name of the machine running the Email DM.

Example 6-2 The System Contains Multiple Oracle DMs Connected to One Database

When your system contains multiple Oracle DMs connected to one database, if not already performed during the BRM server installation, you must initialize the database and drop the tablespaces. You can perform this by modifying each DM machine's **pin_setup.values** file as shown in Table 6-2.

> **Note**
>
> You can initialize the database and drop the tablespaces only *once*.

Table 6-2 DM Machine's pin_setup Values

Entry	Description
$SETUP_INIT_DB	Must be set to **Yes** on the primary Oracle DM machine; on all other machines containing an Oracle DM, set to **No**.
$SETUP_DROP_ALL_TABLES	Must be set to **Yes** on the primary Oracle DM machine; on all other machines containing an Oracle DM, set to **No**.

Example 6-3 Invoices Are Stored in a Separate Database Schema

Storing invoices in their own database schemas speeds up the invoicing process; enables you to store a large number of invoices; and enables you to view, email, and print invoices without affecting the performance of the main BRM schema.

If you are storing invoices in their own schemas, you must update the **pin_setup.values** file to include the correct pointers to the invoice schema. Therefore, you must modify the parameters listed in Table 6-3 on the machine containing the Invoice DM.

> **Note**
>
> The Invoice DM is supported only on Oracle databases.

Table 6-3 Invoice DM Machine's pin_setup Values

Entry	Description
$INVOICE_DB{'user'}	Must contain the user name to log in to your invoice schema. This user name must be different from the one used for the main BRM schema.
$INVOICE_DB{'alias'}	Must contain the database alias of the invoice schema.

Table 6-3   (Cont.) Invoice DM Machine's pin_setup Values

Entry	Description
$INVOICE_DB{'Host'}	Must contain the host name of the machine running the invoice schema.
$INVOICE_DB{'tables_group'}	Must contain the name of the data tablespace.
$INVOICE_DB{'indexes_group'}	Must contain the name of the index tablespace.
$DM_INVOICE{'port'}	Must contain the port number of the Invoice DM. For guidelines, see "Guidelines for Database and Port-Number Entries" in *BRM System Administrator's Guide*.
$DM_INVOICE{'db_num'}	Must contain the database number for the invoice schema.

Running the pin_setup Script

The **pin_setup** script reads the **pin_setup.values** and **pin_tables.values** files and configures BRM by:

- Configuring your various configuration (**pin.conf**) files.
- Setting up database tables and indexes.

To run the **pin_setup** script:

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

   ```
   % ./pin_setup
   ```

2. If you install a DM component on a separate machine from the CM component, you receive the following notification:

   ```
   Warning: File not found: BRM_home/sys/cm/pin.conf
            To complete the install, append the following
            file to the sys/cm/pin.conf file and then restart the CM
            process:
            "BRM_home/append_to_cm_pin_conf"
   ```

   If you receive this notification:

   a. Append the lines from the *BRM_home***/append_to_cm_pin_conf** file to the CM machine's *BRM_home***/sys/cm/pin.conf** file.

   b. Stop and restart the CM process.

3. Check the **pin_setup.log** file for status and errors.

> **Note**
>
> After you run **pin_setup**, the **cm.pinlog** file erroneously contains several PIN_ERROC_FLIST and PIN_ERRCLASS_SYSTEM_DETERMINATE error messages. You can safely ignore these messages.

By default, the **pin_setup** script configures only the last installed product by reading the **pin_setup.values** file. You can also do the following:

- Use the **-all** parameter to configure BRM and all the optional components present in the @COMPONENTS section of **pin_setup.values**. Before using the **-all** parameter to configure BRM, ensure that the following entry is commented:

  ```
  #@COMPONENTS
  ```

For example, go to the *BRM_home*/**setup** directory and enter the following command:

```
pin_setup -all
```

To configure an additional set of components using the **-all** parameter, add the **@COMPONENT_LIST= (" ");** entry in the **pin_setup.values** file.

Changing Your Database Partitions

If you *did not* enable partitioning during installation, skip this section.

If you *did* enable partitioning for one or more storable classes during installation, the tables for those storable classes are now divided into the following partitions:

- **Event tables**
  - If you chose to add 12 fixed partitions to your event tables, those tables were divided into 12 monthly partitions, a historic partition, and a last partition. See "About the Default Partitioning Scheme" in *BRM System Administrator's Guide*.
  - If you chose *not* to add 12 fixed partitions to your event tables, those tables were divided into a historic partition and a last partition (which stores *all* purgeable objects for the table). This partitioning scheme is sufficient for a test or demonstration system. For a production system, however, you must create purgeable partitions.
- **Item tables**

  Tables for item storable classes were divided into a historic partition and a last partition.
- **Tables for all other storable classes in @CLASSES_TO_BE_PARTITIONED**

  Tables for all storable classes listed in the **pin_setup.values** file's @CLASSES_TO_BE_PARTITIONED entry *except* the item storable class were given only one partition, **partition_last**. See "About Objects Stored in partition_last and partition_last_pin" in *BRM System Administrator's Guide*. For a production system, you must create purgeable partitions.

> **Note**
>
> - To test partitioning, you must add at least one purgeable partition.
> - To change your partitioning scheme (for example, by adding purgeable partitions), do so *before* any objects are stored in the partitions you want to modify. *Do not modify a partition after BRM adds objects to the partition.*

For information about changing your partitioning scheme, see "Partitioning Tables" in *BRM System Administrator's Guide*.

Setting Environment Variables for Pipeline Manager

If you installed BRM server and Pipeline Manager on separate machines or directories, on the machine on which you installed the BRM server, set the environment variables to point to the client wallet and JAR files in the *Pipeline_home* directory (where *Pipeline_home* is the directory in which Pipeline Manager is installed) by running the following command:

```
setenv BRM_WALLET Pipeline_home/wallet/client
setenv PIN_HOME Pipeline_home/jars
```

Updating Client Wallet for Pipeline Manager

If you installed BRM server and Pipeline Manager on separate machines, on the machine on which you installed the BRM server, update the values for the following configuration entries in the client wallet manually:

- **oracle.security.client.connect_string***n*
- **oracle.security.client.username***n*
- **oracle.security.client.password***n*

where *n* is the database or database schema number. For example: **oracle.security.client.connect_string1**.

To update configuration entries in the client wallet, see "About Oracle Wallet" in *BRM System Administrator's Guide*.

Copying Files for Compiling CM and FM Modules

To copy the required files for compiling CM and FM modules:

1. Copy the following files from the *BRM_home*/**lib** directory to the *BRM_home*/**PortalDevKit/source/sys/cm** directory:
   - **libcmpin.so**
   - **libdmpin.so**
2. Copy the **cm.cpp** file from the *BRM_home*/**source/sys/cm** directory to the *BRM_home*/**PortalDevKit/source/sys/cm** directory.
3. Copy the contents from the *BRM_home*/**PortalDevKit/include** directory to the *BRM_home*/**include** directory.
4. Open the *BRM_home*/**PortalDevKit/source/samples/env.unix** file in a text editor.
5. 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.
6. Save and close the file.

Editing and Loading the Event Notification List

To edit and load the event notification list:

1. Open the BRM event notification file (*BRM_home*/**sys/data/config/pin_notify**) in a text editor.
2. Search for the following entries and remove them:

```
2354 0 /event/billing/settlement/notify
2355 0 /event/billing/dispute/notify
```

3. Save and close the file.
4. Load the updated file into the database by running the **load_pin_notify** utility:

   ```
   load_pin_notify -v event_notification_file
   ```

   where *event_notification_file* is the path to the BRM event notification file.

   See "Loading the Event Notification List" in *BRM Developer's Guide* for more information.

Loading the Tailor-Made Stored Procedure

If you use the Tailor-Made Plan feature, you must load its stored procedure.

To load the stored procedure:

1. Ensure the following:
   - BRM and Pipeline Manager are installed.
   - The BRM schema and the Pipeline Manager schema reside on the same database.
2. Connect to the Oracle database with SQL*Plus:

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

3. Grant access of the pipeline schema to the BRM database user (for example, **pin**) by doing the following:
   a. Grant permissions to select, update, insert, and delete command on the Pipeline Manager tables listed below. For example:

      ```
      SQL> grant select, update, insert, delete on tableName to pin_user;
      ```

      where *tableName* is the name of the Pipeline Manager table and *pin-user* is the BRM database user. Run the command for the following tables:
      - ifw_rateplan
      - ifw_rateplan_cnf
      - ifw_rateplan_ver
      - ifw_model_selector
      - ifw_selector_detail
      - ifw_selector_rule
      - ifw_selector_rule_lnk
      - ifw_selector_ruleset
      - ifw_pricemodel
      - ifw_pricemdl_step

   b. Grant select permission on the Pipeline Manager tables and sequences listed below. For example:

      ```
      SQL> grant select on tableOrSequenceName to pin_user;
      ```

where *tableOrSequenceName* is the name of the Pipeline Manager table or sequence and *pin-user* is the BRM database user. Run the command on the following tables and sequences:

- ifw_service
- ifw_timezone
- ifw_timemodel
- ifw_impact_cat
- ifw_zonemodel
- ifw_calendar
- ifw_seq_selectordetail
- ifw_seq_selectorrule
- ifw_seq_modelselector
- ifw_seq_pricemodel
- ifw_seq_rateplan

4. Type **exit** to exit SQL*Plus.
5. Go to the *Pipeline_home***/database/Oracle/Scripts** directory, where *Pipeline_home* is the directory in which Pipeline Manager is installed.
6. Enter the following command to open SQL*Plus:

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

   where *pin_user* is the BRM database user and *databaseAlias* is the service name or database alias of the Oracle database.
7. Enter the following command to load the stored procedure:

   ```
   SQL>@create_pricing_tailormadeplan_procedures.plb
   ```
8. Type **exit** to exit SQL*Plus.

Postinstallation Tasks for a BRM Production Installation

After installing the BRM server, if you have not performed the following tasks during installation, do the following and then run the **pin_setup** script. See "Running the pin_setup Script" for more information.

- (Optional) To further configure BRM, you can edit the *BRM_home***/setup/pin_setup.values** file. See "Editing the pin_setup.values File" for more information.
- (Optional) If you did not create multiple tablespaces during the BRM server installation, create additional tablespaces and map the BRM tables to the tablespaces. See "Editing the pin_tables.values File" for more information.
- (Optional) If you enabled partitioning during BRM installation, the following classes are automatically partitioned: journal, journal master, newsfeed, and user activity. However, you can change the classes that are partitioned. See "Enabling Different Classes for Partitioning during Installation" for more information. This task is common to both typical and complete BRM server installations.

Do the following *after* running the **pin_setup** script:

- (Optional) Change your database partitions as required. See "Changing Your Database Partitions" for more information.
- If you installed the BRM server and Pipeline Manager on separate machines or directories, set the BRM_WALLET and PIN_HOME variables in the BRM server to point to the client wallet and JAR files in the *Pipeline_home* directory, where *Pipeline_home* is the directory in which Pipeline Manager is installed. See "Setting Environment Variables for Pipeline Manager" for more information.
- If you have installed the BRM server and Pipeline Manager on separate machines, update the configuration entries for Pipeline Manager in the client wallet manually. See "Updating Client Wallet for Pipeline Manager" for more information.
- If you installed any optional components on a separate machine from the CM, edit the optional component's configuration entries. See "Editing Optional Components Configuration Entries" for more information.
- Copy the makefiles required for compiling the Connection Manager instances. See "Copying Files for Compiling CM and FM Modules" for more information.
- Edit and load the event notification list. See "Editing and Loading the Event Notification List" for more information.
- Load the tailor-made stored procedure to create a tailor-made plan. See "Loading the Tailor-Made Stored Procedure" for more information.

Editing the pin_tables.values File

If you install BRM with the default settings, BRM data is stored in a single tablespace (**PIN_TABLE**), and BRM indexes are stored in a second tablespace (**PIN_INDEX**). If you create these tables with the **bigfile** option, you do not need more tablespaces.

If, however, your BRM database contains more than two tablespaces, you must use the *BRM_home***/setup/scripts/pin_tables.values** file to map the BRM tables to the tablespaces you created in the database.

Enabling Different Classes for Partitioning during Installation

To enable partitioning for different storable classes, you must edit the **pin_setup.values** file and then run the **pin_setup** script.

To enable different storable classes for partitioning during installation:

1. Open the *BRM_home***/setup/pin_setup.values** file.
2. Ensure that the $ENABLE_PARTITION parameter is set to **Yes**.
3. In the @CLASSES_TO_BE_PARTITIONED entry, add or remove storable classes. The default list is shown below:

   ```
   @CLASSES_TO_BE_PARTITIONED = ("/journal:local","/journal master:local", "/
   newsfeed:local","/sepa:local","/user_activity:local");
   ```

   To add classes for partitioning, use the following format:

   ```
   "/class_name:index_type"
   ```

   where:

   - *class_name* is the name of a base storable class. Add only base storable classes to the list.

The event storable class is never added to the list. If partitioning is enabled ($ENABLE_PARTITION = **Yes**), the event tables are always partitioned.

- *index_type* is one of the following index types available for non-event partitions (all event indexes are local):
  - **:local** — Maintenance can be done without shutting down BRM, but performance may suffer. Search operations are more time consuming because searches must hit every local index.
  - **:global** — Search operations are fast, but BRM services must be shut down before adding or dropping partitions. Tables with global indexes require the indexes to be rebuilt.

> ⓘ **Note**
>
> When **pin_setup** is run, the tables of storable classes enabled for partitioning get an initial partition. Because BRM does not provide a tool to revert partitioned tables to nonpartitioned tables, these partitions cannot easily be removed after data starts being stored in them.

4. Save and close the file.

Editing Optional Components Configuration Entries

If you install any optional components on a separate machine from your CM, manually edit the component's configuration entries to include the following:

- The correct CM port number.
- The correct host name for the CM machine.
- Any component-specific Facilities Module (FM) entries. For information, see "Syntax for Facilities Module Entries" in *BRM System Administrator's Guide*.

For updating the configuration entries in the Oracle wallet, see "About Oracle Wallet" in *BRM System Administrator's Guide*.

Maintaining a BRM Database

You monitor and maintain the BRM database with standard database tools. For example, you can set up your database software to generate log files. For more information, see the documentation for your database software.

You can also use the **sar** utility to monitor performance.

> ⓘ **Note**
>
> Do not use SQL statements to insert, delete, or update BRM tables or objects. Always use the Portal Communications Module (PCM) interface, which guarantees the integrity of the BRM database.

For information about managing a multischema system, see "Managing a Multischema System" in *BRM System Administrator's Guide*.

Maintaining the Connection to the Database

If the connection to the database fails, BRM automatically attempts to reconnect using the database name listed in the DM configuration file. If BRM can reestablish the connection, BRM generally restarts the operation. If BRM is in the middle of a transaction, BRM reports a **PIN_ERR_STORAGE** error in the DM log file. If BRM cannot reestablish the connection, it reports a **PIN_ERR_STORAGE_DISCONNECT** error.

For more information about failure recovery, see "Four-Tier Architecture and Failure Recovery" in *BRM Concepts*.

Monitoring Database Space

Before you installed BRM, you set up your database based on estimates of the size of tables for your business activity. The planning process also included forecasts of how fast the tables would grow. You should monitor the growth of tables not only to ensure you maintain enough space on your system, but also to check for unexpected growth that would indicate some problem.

On a typical production system, you should check tables monthly. As part of this audit, you should match rows in each of the tables against the expected rows. If you spot discrepancies, checking the individual tables shows where the unexpected growth is coming from. For a list of the BRM tables, see "Storable Class-to-SQL Mapping" in *BRM Developer's Reference*.

If you have a multischema system, you can use growth information to revise your scheme for distributing accounts among your various schemas. See "Setting Database Schema Priorities" in *BRM System Administrator's Guide*.

Monitoring Oracle Tablespace Usage

You should monitor the growth of tables so that you can add more extents or data files before the tablespaces are filled. To find information about the available blocks and bytes for each tablespace, run the following command:

```
SQL> select * from user_free_space;
```

Your database administrator should provide a maximum value to look for. If a tablespace grows past that maximum, you (or an automated script) should notify the database administrator to take appropriate action.

Monitoring SQL Statements

You can collect debugging information by gathering the SQL statements generated by **dm_oracle** processes. The statements appear in the DM log file, not the DM **pinlog** file.

To get the SQL statements for a specific operation or sequence of events:

1. In the environment from which **dm_oracle** will be started, set the environment variable **DM_DEBUG3** to **0xFFFF003F**.

   Using the c-shell (**csh**):

   ```
   setenv DM_DEBUG3 0xFFFF003F
   ```

   Using the bash or Korn shell (**sh/ksh**):

   ```
   export DM_DEBUG3=0xFFFF003F
   ```

2. Clear the old log file.

3. Start the Oracle DM.
4. Run the DM operation you are debugging to generate SQL statements.
5. Stop the Oracle DM.
6. Use the **grep** command on the Oracle DM log file for the "SQL_STMT" string.
7. Unset the **DM_DEBUG3** environment variable. Otherwise, subsequent DM operations will generate huge log files.

   Using the c-shell (**csh**):

   ```
   unsetenv DM_DEBUG3
   ```

   Using the bash or Korn shell (**sh/ksh**):

   ```
   unset DM_DEBUG3
   ```

Rebuilding Indexes

The structure of indexes influences the speed at which BRM can find records in the database. While you are using BRM in a production environment, especially when there is intensive inserting in the database, these indexes can become unbalanced, impeding access to BRM records. For best efficiency, rebuild the indexes frequently. For example, if you have a heavily used production system, you might want to rebuild the indexes weekly.

> **Note**
>
> Do *not* delete any of the standard BRM indexes without first consulting Oracle. Removing an index can lead to serious performance problems. Also, do *not* delete or change any of the standard stored procedures. Otherwise, the DM might malfunction.

Configuring Views for Oracle Application Infrastructure Architecture

In single-schema or multischema environments, you must create or configure a unified collections action view and a unified collections scenario view to be queried by Oracle Application Infrastructure Architecture (Oracle AIA).

To configure the views for Oracle AIA:

1. On the primary database schema, run the following commands, which grant permissions for each of the secondary schemas:

   > **Note**
   >
   > This step is not required for single-schema systems.

   ```
   sqlplus login/password@database_alias
   SQL>GRANT SELECT ON CONFIG_COLLECTIONS_ACTION_T TO schema_name WITH GRANT OPTION;
   SQL>GRANT SELECT ON CONFIG_COLLECTIONS_SCENARIO_T TO schema_name WITH GRANT OPTION;
   ```

   where:

   - *login* is the username for the primary database schema.

- *password* is the password for *login*.
- *database_alias* is the BRM database alias of the primary schema.
- *schema_name* is the name of the secondary schema.

2. On each secondary database schema, run the following commands, which grant permissions on tables required for collections action and scenario view:

   > **Note**
   >
   > This step is not required for single-schema systems.

   ```
   sqlplus login/password@database_alias
   SQL>GRANT SELECT ON AU_COLLECTIONS_ACTION_T TO schema_name;
   SQL>GRANT SELECT ON COLLECTIONS_SCENARIO_T TO schema_name;
   SQL>GRANT SELECT ON BILLINFO_T TO schema_name;
   SQL>GRANT SELECT ON COLL_ACTION_IF_VIEW To schema_name;
   SQL>GRANT SELECT ON COLL_SCENARIO_IF_VIEW to schema_name;
   SQL>exit
   ```

   where:

   - *login* is the username for the secondary database schema.
   - *password* is the password for *login*.
   - *database_alias* is the BRM database alias of the secondary schema.
   - *schema_name* is the name of the primary schema.

3. On the primary database schema, run the following command:

   ```
   sqlplus login/password@database_alias
   ```

   where:

   - *login* is the username for the primary database schema.
   - *password* is the password for *login*.
   - *database_alias* is the BRM database alias of the primary schema.

4. Do the following:

   - To create a unified collections action view, run the following command:

     ```
     SQL>CREATE OR REPLACE VIEW "UNIFIED_COLL_ACTION_IF_VIEW" AS SELECT * FROM
         schema1.COLL_ACTION_IF_VIEW UNION SELECT * FROM schema2.COLL_ACTION_IF_VIEW;
     ```

   - To create a unified collections scenario view, run the following command:

     ```
     SQL>CREATE OR REPLACE VIEW "UNIFIED_COLL_ACTION_IF_VIEW" AS SELECT * FROM
         schema1.COLL_ACTION_IF_VIEW UNION SELECT * FROM schema2.COLL_ACTION_IF_VIEW;
     ```

     where:

     - *schema1* is the name of the primary schema.
     - *schema2* is the name of each of the secondary schemas.

5. Run the following command, which exits SQL*Plus:

   ```
   SQL>exit
   ```

What's Next?

After completing the postinstallation tasks, set up your product offerings, business policies, and customer accounts in BRM. To do so, you must install the BRM client applications, including the BRM web-based client applications, such as Billing Care, Business Operations Center, and Pricing Design Center (PDC):

- To set up BRM, install core BRM client applications. See "Installing BRM Thick Clients" for more information.
- To create and manage customer accounts, install the Billing Care application. See *Billing Care Installation Guide* for more information.
- To create, schedule, and view the results of business operations, install the Business Operations Center application. See *Business Operations Center Installation Guide* for more information.
- To create product offerings, install PDC. See *PDC Installation Guide* for more information.

For a production installation, to implement BRM with the required software products for charging, billing, and revenue management, you must install and configure the required software products before you implement BRM.

See the product documentation of the required software products for instructions on installing and configuring those products. For example, if you are using Elastic Charging Engine (ECE) for usage charging, install ECE. See *ECE Installation Guide* for more information.

7 Installing a Multischema System

Learn how to install Oracle Communications Billing and Revenue Management (BRM) in a multischema system.

Topics in this document:

- Before Installing a Multischema System
- Multischema Installation Overview
- Installing a Multischema System: Task List
- What's Next?

Before Installing a Multischema System

Before you install a multischema system, verify the following:

- Your system meets the minimum hardware and software requirements.

  See "System Requirements".
- You have all the software required to run BRM with an Oracle database in an Oracle Real Application Clusters (Oracle RAC) system.

  See "Database Requirements".
- All database schemas in your system use the following:
  - A unique schema name
  - The same tablespace names

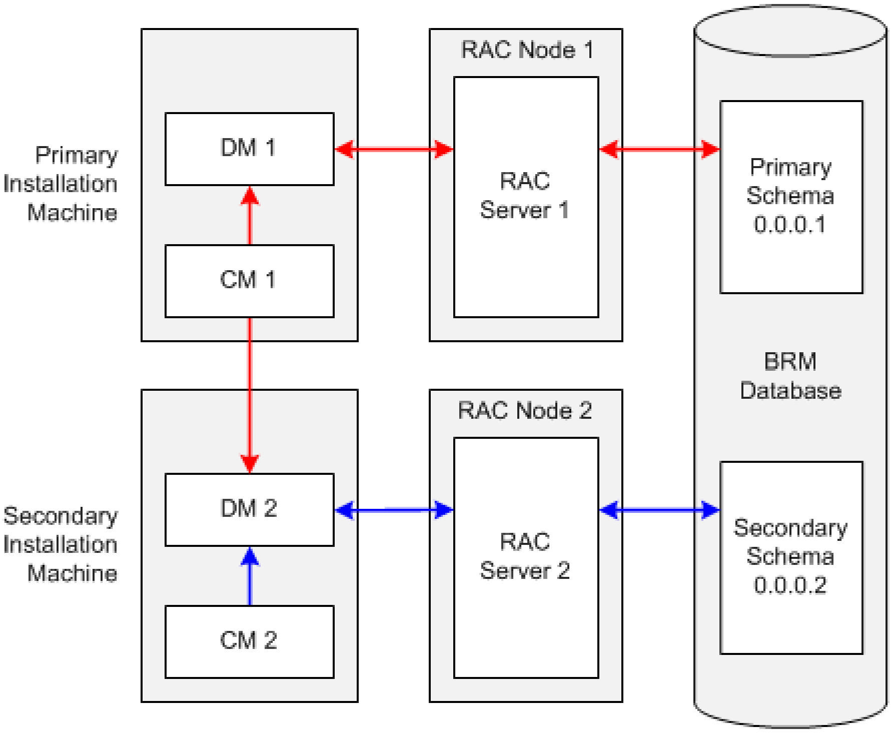
Multischema Installation Overview

You can distribute BRM data among multiple database schemas to increase scalability and availability. For more information about multischema systems, see "A BRM Multischema Production System".

Typically, a multischema system includes one or more database schema machines and several machines running BRM. You configure the additional BRM machines so that each Oracle Data Manager (DM) process communicates with one particular schema across the network.

For example, Figure 7-1 shows a multischema system with two BRM installation machines and two schemas.

Figure 7-1   Two-Schema BRM Multischema Environment



In a multischema system:

- The *primary installation machine* must contain, at a minimum, the primary Connection Manager (CM), the primary Oracle DM, and a database client. The primary CM communicates with both primary and secondary Oracle DMs, and the primary DM is dedicated to the primary database schema.
- The *secondary installation machine* must contain, at a minimum, a secondary DM and a database client. In Figure 7-1, it also contains a secondary CM. The secondary CM communicates with the secondary DM, and the secondary DM is dedicated to the secondary database schema. Although the secondary CM is not required, it enables you to perform dedicated operations, such as billing, on the secondary database schema.
- The *primary database schema* contains all the configuration, pricing, audit trail, and uniqueness objects. Modifications to configuration, pricing, and audit trail objects are always performed by the primary DM in the primary database schema. Secondary DMs can modify uniqueness objects in the primary database schema by using schema qualifications. The primary database schema also contains subscriber data, such as events and account objects.
- The *secondary database schema* contains subscriber data. Each secondary database schema also has access to the configuration, pricing, audit trail, and uniqueness objects stored in the primary database schema.

Installing a Multischema System: Task List

This section provides an overview of the tasks you must perform to install and configure a multischema system.

1. Install an Oracle database across all the servers in your Oracle RAC system.

For more information, see the following:

- BRM Preinstallation Tasks
- Setting Up Oracle RAC
- Oracle RAC documentation

2. Through one Oracle RAC node, create the primary database schema.

   See "Creating and Configuring pin_user".

   > ⓘ **Note**
   >
   > The term "BRM user" is synonymous with the term "database schema."

3. Through each of the other Oracle RAC nodes, create a secondary database schema.

   See "Creating and Configuring pin_user".

4. Configure each database schema in your system to connect to all the other database schemas, and verify that the connections work.

   > ⓘ **Note**
   >
   > This enables the database links created when you run the installer to work across the schemas.

   For information, see your Oracle database documentation.

5. For each database schema, grant execute permission for **dbms_lock**.

   See "Creating and Configuring pin_user".

6. On your primary installation system, install the PERL libraries and the JRE required to install BRM components.

7. On each of your secondary installation systems, install the PERL libraries and the JRE required to install BRM components.

8. On your primary installation system, install BRM.

   See "Installing BRM on the Primary Installation Machine".

9. On your secondary installation systems, install BRM.

   See "Installing BRM on a Secondary Installation Machine".

10. Verify that the BRM installation systems can connect to all the schemas.

    See "Verifying That the Installation Machines Connect to All Schemas".

11. Install Multidatabase Manager on the primary installation system.

    See "Installing and Configuring Multidatabase Manager on the Primary Installation Machine".

12. (Optional) Disable the creation of uniqueness objects.

    See "Disabling Creation of Uniqueness Objects".

13. Edit the **pin_multidb.conf** file on the primary installation system.

    See "Configuring pin_multidb.conf on the Primary Installation Machine".

14. Edit the client wallet on the primary installation system.

    See "Configuring Client Wallet on the Primary Installation Machine".

15. If your system includes an SSL-enabled Oracle database, edit the SSL database wallet on the primary installation system. See "Connecting to SSL-Enabled Oracle Database".

16. Complete the configuration on the primary system.

    See "Completing Configuration of the Primary Installation System".

17. Complete the configuration on the secondary system.

    See "Completing Configuration of the Secondary Installation System".

18. (Optional) Make custom tables available to all the schemas in your system.

    See "Creating Custom Tables That Are Available to All Database Schemas" in *BRM System Administrator's Guide*.

19. (Optional) Configure BRM to enforce the creation of unique account numbers.

    See "Enforcing the Creation of Unique Account Numbers".

20. Create schema qualifications that connect each secondary database schema to the primary database schema.

    See "Setting Up Schema Qualifications".

21. Update the views for multischema systems. See "Updating Views for Multischema Systems".

Installing BRM on the Primary Installation Machine

The primary installation machine must contain, at a minimum, the primary CM, the primary Oracle DM, and a database client.

> **Note**
>
> The database client is required because the primary Oracle DM and the primary database schema are installed on different machines.

To set up the primary installation machine:

1. Install BRM. See "Installing BRM".
2. Connect the Oracle DM to the primary Oracle RAC instance:
    a. Open the Oracle DM configuration file (*BRM_home*/**sys/dm_oracle/pin.conf**) in a text editor.
    b. Set the **sm_database** entry to the connect descriptor configured for the primary Oracle RAC instance:

    ```
    - dm sm_database connectionString
    ```

    connectionString must match the *connectionString* entry for the primary Oracle RAC instance in the **tnsnames.ora** file.

   c. Add the following entry:

      ```
      - dm sm_svcname serviceName
      ```

      serviceName must match the **SERVICE_NAME** entry in the connect descriptor specified in the preceding step.

   d. Save and close the file.

   e. Stop and restart the Oracle DM.

3. Configure the primary database client to connect to each secondary database schema.

   For information, see "Installing the Database and Oracle DM on Separate Machines".

4. Install the optional components you purchased that might add tables to your database.

Installing BRM on a Secondary Installation Machine

A secondary installation machine must contain, at a minimum, an Oracle DM and a database client.

> **Note**
>
> - The database client is required because a secondary installation machine and its corresponding secondary database schema machine (which is in an Oracle RAC system) are different machines.
> - You must install all of the BRM software on all machines that run a BRM process, even if you run only a CM or a DM on a particular machine.

To set up a secondary installation machine:

1. Copy the BRM wallet from your primary installation machine to your secondary installation machine.

2. Set the BRM_WALLET and BRM_CONF_WALLET environment variables to the location of your BRM wallet on the secondary installation machine:

   ```
   setenv BRM_WALLET BRM_home/wallet/client
   setenv BRM_CONF_WALLET BRM_home/wallet/client
   ```

3. Install BRM. See "Installing BRM".

4. Connect the Oracle DM to a secondary Oracle RAC instance:

   a. Open the Oracle DM configuration file (*BRM_home***/sys/dm_oracle/pin.conf**) in a text editor.

   b. Set the **sm_database** entry to the connect descriptor configured for a secondary Oracle RAC instance:

      ```
      - dm sm_database connectionString
      ```

      connectionString must match the *connectionString* entry for a secondary Oracle RAC instance in the **tnsnames.ora** file.

   c. Add the following entry:

      ```
      - dm sm_svcname serviceName
      ```

serviceName must match the **SERVICE_NAME** entry in the connect descriptor specified in the preceding step.

d. Save and close the file.

e. Stop and restart the Oracle DM.

5. Configure the secondary database client to connect to the primary database schema.

   For information, see "Installing the Database and Oracle DM on Separate Machines".

6. Install the same optional components that you installed on the primary installation machine.

7. On the primary installation machine, generate the root key by running the following command:

   ```
   pin_crypt_app -genrootkey
   ```

   The root key is generated.

8. On both primary and secondary installation machines, set the generated primary root key in the Oracle DM configuration file (*BRM_home***/sys/dm_oracle/pin.conf**).

Verifying That the Installation Machines Connect to All Schemas

Verify that BRM was installed and set up properly on your installation machines by connecting to the primary and secondary database schemas.

Using testnap to Verify Access to Your Schemas

The BRM **testnap** utility tests your installation machine's connection to the database schema by establishing a Portal Communications Module (PCM) connection with the CM and executing PCM opcodes using that connection. For more information, see "testnap" in *BRM Developer's Reference*.

Verifying Access between Primary Installation Machine and Primary Schema

Perform the following procedure on the primary installation machine to verify that it can connect to the primary database schema:

1. Log in as user **pin**, go to the *BRM_home***/sys/test** directory, and open the **pin.conf** file in a text editor such as **vi**:

   ```
   su - pin
   cd BRM_home/sys/test
   vi pin.conf
   ```

2. Modify the following entries in the **pin.conf** file:

   ```
   - nap cm_ptr ip primary_hostname port_number
   - nap login_name login_name
   ```

   where:

   - *primary_hostname* is the host name of the primary installation machine.
   - *port_number* is the port number of the primary CM.
   - *login_name* is the login for the primary CM (the default is **root.0.0.0.1**).

3. Run the **testnap** utility from the *BRM_home***/sys/test** directory, and check for connection errors:

```
testnap
====>database 0.0.0.1 from pin.conf "userid"
q
```

Verifying Access between Secondary Installation Machines and Secondary Schemas

Perform the following procedure on each secondary installation machine to verify that it can connect to its secondary database schema:

1. Log in as user **pin**, go to the *BRM_home***/sys/test** directory, and open the **pin.conf** file in a text editor such as **vi**:

   ```
   cd BRM_home/sys/test
   vi pin.conf
   ```

2. Modify the following entries in the **pin.conf** file:

   ```
   - nap cm_ptr ip secondary_hostname port_number
   - nap login_name login_name
   ```

   where:

   - *secondary_hostname* is the host name of the secondary installation machine.
   - *port_number* is the port number of the secondary CM.
   - *login_name* is the login for the secondary CM (the default is **root.0.0.0.2**).

3. Run the **testnap** utility from the *BRM_home***/sys/test** directory, and check for connection errors:

   ```
   testnap
   ====>database 0.0.0.2 from pin.conf "userid"
   q
   ```

Installing and Configuring Multidatabase Manager on the Primary Installation Machine

To install and configure Multidatabase Manager, perform the following procedure on the primary installation machine:

1. Stop all BRM processes if running. See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.
2. Install Multidatabase Manager. See "Installing Individual BRM Components".
3. Go to the *BRM_home***/setup** directory, and run the **pin_setup** script.
4. Log in as user **pin** and copy the Multidatabase Manager configuration file (*BRM_home***/apps/multi_db/pin.conf**) to the *BRM_home***/setup/scripts** directory:

   ```
   su - pin
   cp BRM_home/apps/multi_db/pin.conf BRM_home/setup/scripts
   ```

5. Go to the *BRM_home***/setup/scripts** directory, and open the **pin.conf** file in a text editor such as **vi**:

   ```
   cd BRM_home/setup/scripts
   vi pin.conf
   ```

6. Modify the following entries in the **pin.conf** file:

   ```
   - nap cm_ptr ip primary_hostname port_number
   - nap login_name login_name
   ```

where:

- *primary_hostname* is the host name of the primary installation machine.
- *port_number* is the port number of the primary CM.
- *login_name* is the login for the primary CM (the default is **root.0.0.0.1**).

7. Start the CM and Oracle DM processes.

   See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

8. Run the **testnap** utility from the *BRM_home***/setup/scripts** directory, and check for connection errors:

   ```
   testnap
   ====>database 0.0.0.1 from pin.conf "userid"
   q
   ```

Disabling Creation of Uniqueness Objects

BRM requires unique service login names. The **/uniqueness** objects maintain a list of unique login names across multiple database schemas. The list is used during authentication. You can disable creation of **/uniqueness** objects. For example, you might want to disable creation of **/uniqueness** objects if you use your own authentication software.

To disable creation of **/uniqueness** objects:

1. Open the CM configuration file (*BRM_home***/sys/cm/pin.conf**).
2. Edit the following entry:

   ```
   -cm uniqueness_login 1
   ```

   where:

   - **0** does not create uniqueness objects.
   - **1** creates uniqueness objects. This is the default.

3. Save the file.
4. Stop and restart the CM.

Configuring pin_multidb.conf on the Primary Installation Machine

You must enter configuration parameters for your multischema setup in the **pin_multidb.conf** file.

> **Note**
>
> This file is accessible only if you installed Multidatabase Manager for the appropriate platform. See "Installing and Configuring Multidatabase Manager on the Primary Installation Machine".

To configure a BRM multischema system:

1. Open the *BRM_home***/setup/scripts/pin_multidb.conf** file in a text editor such as **vi**:

   ```
   vi BRM_home/setup/scripts/pin_multidb.conf
   ```

2. Modify the configuration entries listed in Table 7-1.

> **Note**
>
> Include a set of **$PIN_MD_SECONDARY*[*x*]** entries for each secondary database schema in your system. For example, if your system contains three secondary database schemas, the file should contain a set of **$PIN_MD_SECONDARY*** entries ending in **[0]**, a set ending in **[1]**, and a set ending in **[2]**.
>
> In prior versions, you needed to set the **$PIN_MD_PRIMARY_DBNO**, **$PIN_MD_PRIMARY_OWNER**, **$PIN_MD_PRIMARY_DBNAME**, **$PIN_MD_PRIMARY_HOSTNAME**, and **$PIN_MD_PRIMARY_PORT** parameters. It is no longer necessary to set them, because they are set automatically.

Table 7-1 pin_multidb.conf File Entries

Entry	Description
$JDBC_JAR_FILE	Specify the location of the JDBC Library (**ojdbc11.jar**).
$PIN_MD_SQL_PLATFORM	Specify the database software your multischema system uses. Enter **'Oracle'**.
$PIN_MD_SQL_BASE	Set this to **$ORACLE_HOME**.
$PIN_MD_INFRANET_BASE	Specify the directory in which you installed BRM on your primary installation machine. You must change this value if you installed BRM in a directory other than the default directory.
$PIN_MD_CM_HOST	Enter the machine name where the primary CM is running.
$PIN_MD_CM_PORT	Enter the port number for the primary CM. The default is **11960**. Check the port number by looking in the *BRM_home***/sys/cm/pin.conf** file.
$PIN_MD_SECONDARY_START_INST	Specify the **$PIN_MD_SECONDARY*[*x*]** array number for the first secondary database schema to add to your system, where *x* is the array number. For example: • When initially setting up a multischema system containing three secondary database schemas, set this entry to **"0"** and set **$PIN_MD_SECONDARY_STOP_INST** to **"2"**. This tells the installer to configure secondary database schemas based on arrays 0, 1, and 2. • When adding secondary database schemas to an existing multischema system, enter the **$PIN_MD_SECONDARY*[*x*]** array number for the first schema being added to the system. For example, if your system contains three secondary database schemas (0, 1, and 2) and you are adding two more schemas, set this entry to **"3"** and set **$PIN_MD_SECONDARY_STOP_INST** to **"4"**. This tells the installer to configure secondary database schemas based only on arrays 3 and 4 and to ignore arrays 0, 1, and 2. **Note:** You must include a set of **$PIN_MD_SECONDARY*[*x*]** entries for each secondary database schema in your system. For example, if your system contains three secondary database schemas, the **pin_multidb.conf** file should contain a set of **$PIN_MD_SECONDARY*** entries ending in **[0]**, a set ending in **[1]**, and a set ending in **[2]**. The default is **"0"**.

Table 7-1 (Cont.) pin_multidb.conf File Entries

Entry	Description
$PIN_MD_SECONDARY_END_INST	Specify the **$PIN_MD_SECONDARY*[*x*]** array number for the last secondary database schema to add to your system, where *x* is the array number. For example, when you initially set up a multischema system containing three secondary database schemas, enter **"2"**. When adding secondary database schemas to an existing multischema system, enter the **$PIN_MD_SECONDARY*[*x*]** array number for the last schema being added to the system. For example, if your system contains three secondary database schemas (corresponding to arrays 0, 1, and 2) and you are adding two more schemas, enter **"4"**. The default is **"0"**.
$PIN_MD_SECONDARY_DBNO [*x*]	Enter the secondary database number. The default is **"0.0.0.2"**.
$PIN_MD_SECONDARY_OWNER [*x*]	Enter your user name for the secondary database schema. The default is **"PINB"**.
$PIN_MD_SECONDARY_DBNAME [*x*]	Enter the database alias for the secondary database schema. **Note:** Ensure that you enter a unique database alias for each secondary database schema.
$PIN_MD_SECONDARY_HOSTNAME [*x*]	Enter the machine name where the secondary Oracle DM is running.
$PIN_MD_SECONDARY_PORT [*x*]	Enter the DM port number for the secondary database schema.
$PIN_MD_SECONDARY_PARENT_DBNO [*x*]	Enter the database number of the primary database schema. The default is **"0.0.0.1"**.
$PIN_MD_SECONDARY_PARENT_SCHEMA [*x*]	Enter the name of the primary database schema. This value must match the **$PIN_MD_PRIMARY_OWNER [*x*]** value.

Configuring Client Wallet on the Primary Installation Machine

On your primary installation machine, enter the configuration parameters for your multischema setup in the client wallet by using the **pin_crypt_app** utility. See "pin_crypt_app" in *BRM Developer's Guide* for more information about the utility's syntax and parameters.

To configure the client wallet, perform the following on the primary installation machine for each secondary database schema:

1. Go to the *BRM_home***/bin** directory.
2. Add or modify the values in the client wallet by running the following command:

   ```
   pin_crypt_app -setconf -wallet clientWalletLocation -parameter configEntry -value
   value
   ```

   where:

   - *clientWalletLocation* is the path to the client wallet.
   - *configEntry* is the configuration entry in the client wallet. See Table 7-2 for the entries to be added or modified.
   - *value* is the value of the configuration entry to store in the client wallet.

> **Note**
>
> Ensure that you add or modify the entries listed in Table 7-2 for each secondary schema.

Table 7-2 Client Wallet Entries

Entry	Description
oracle.security.client.connect_string*x* where *x* is the array number of the secondary database schema. For example, if your system contains three secondary database schemas, the client wallet should contain a set of schema entries ending in **[0]**, a set ending in **[1]**, and a set ending in **[2]**.	Enter the database alias for the secondary database schema. **Note:** Ensure that you enter a unique database alias for each secondary database schema.
oracle.security.client.username*x*	Enter your user name for the secondary database schema. The default is **"PINB"**.
oracle.security.client.password*x*	Enter your password for the secondary database schema.
0.0.0.*x***_user_password**	Enter the database password.

The configuration entries are stored in the client wallet.

Connecting to SSL-Enabled Oracle Database

You connect your BRM multischema system to an SSL-enabled Oracle database by updating the SSL database wallet. To do so, you use the **pin_crypt_app** utility to add the following configuration entries to the SSL database wallet:

- **oracle.security.client.connect_string***X*: Specifies the database alias for a secondary database schema.
- **oracle.security.client.username***X*: Specifies the user name for a secondary database schema.
- **oracle.security.client.password***X*: Specifies the password for the secondary database schema.

where *X* is the secondary database schema number. That is, **1** for the first secondary database schema, **2** for the second secondary database, and so on.

For example, if your multischema system contains a primary database schema and two secondary database schemas, you would add the following configuration entries:

```
oracle.security.client.connect_string1
oracle.security.client.username1
oracle.security.client.password1
oracle.security.client.connect_string2
oracle.security.client.username2
oracle.security.client.password2
```

To add configuration entries to the SSL database wallet, do the following:

1. Go to the *BRM_home***/bin** directory.
2. Run the following command for each configuration entry:

   ```
   pin_crypt_app -setconf -wallet DBWalletLocation -parameter configEntry -value value
   ```

   where:

   - *DBWalletLocation* is the path and file name of the SSL database wallet.
   - *configEntry* is the name of the configuration entry.
   - *value* is the value of the configuration entry.

   When the prompt appears, enter the password for the wallet.

Completing Configuration of the Primary Installation System

To complete the configuration of the primary system, perform the following procedure on the primary installation system:

1. Log in as the BRM user, and change the status of all secondary database schemas to **OPEN** in the *BRM_home***/apps/multi_db/config_dist.conf** file.

   > **Note**
   >
   > If your system contains multiple secondary database schemas, create a new set of entries for each additional secondary database schema.

   For example:

   ```
   DB_NO = "0.0.0.1" ;              # 1st database config block.
   PRIORITY = 1 ;
   MAX_ACCOUNT_SIZE = 100000 ;
   STATUS = "OPEN" ;
   SCHEMA_NAME = "pin111x" ;

   DB_NO = "0.0.0.2" ;              # 2nd database config block.
   PRIORITY = 1 ;
   MAX_ACCOUNT_SIZE = 100000 ;
   STATUS = "OPEN" ;
   SCHEMA_NAME = "pin112x" ;
   ```

2. Ensure the following lines are present in the *BRM_home***/sys/cm/pin.conf** file on the primary system:

   ```
   - cm primary_db 0.0.0.1 / 0
   - cm dm_attributes 0.0.0.1 scoped,assign_account_obj,searchable
   - cm dm_pointer 0.0.0.1 ip prim_dm_host prim_dm_port
   ```

   where:

   - *prim_dm_host* is the host name of the primary installation system
   - *prim_dm_port* is the port of the DM for the primary schema

3. Locate or add the following lines for *each* secondary DM in the *BRM_home***/sys/cm/pin.conf** CM configuration file:

```
- cm dm_attributes sec_db_num scoped,assign_account_obj,searchable
- cm dm_pointer sec_db_num ip sec_dm_host sec_dm_port
```

where:

- *sec_db_num* is the database number of the secondary schema, for example **0.0.0.2**
- *sec_dm_host* is the host name of the secondary installation system
- *sec_dm_port* is the port of the secondary DM

> **Note**
>
> Your CM **pin.conf** file must contain **dm_pointer** and **dm_attributes** entries for each secondary DM in your system.

4. Verify that the **pin_config_distribution** process is running. See "pin_config_distribution" in *BRM System Administrator's Guide*.

Completing Configuration of the Secondary Installation System

To complete the configuration of the secondary system, perform the following procedure on *each* secondary installation system:

1. Make the data dictionary objects writable by setting the following entries to **1** the *BRM_home***/sys/dm_oracle/pin.conf** file:

   ```
   - dm dd_write_enable_fields 1
   - dm dd_write_enable_objects 1
   - dm dd_write_enable_portal_objects 1
   ```

2. Locate or add the following entries to the *BRM_home***/sys/cm/pin.conf** file on each secondary CM that needs access to all the schemas in your system.

   ```
   - cm primary_db 0.0.0.1 / 0
   - cm dm_attributes 0.0.0.1 scoped,assign_account_obj,searchable
   - cm dm_pointer 0.0.0.1 ip prim_dm_host prim_dm_port
   - cm dm_pointer sec_db_num ip sec_dm_host sec_dm_port
   ```

   where:

   - *prim_dm_host* is the host name of the primary installation system
   - *prim_dm_port* is the port of the DM for the primary schema
   - *sec_db_num* is the database number of the secondary schema, for example **0.0.0.2**
   - *sec_dm_host* is the host name of the secondary installation system
   - *sec_dm_port* is the port of the secondary DM

   > **Note**
   >
   > Your **pin.conf** file must contain a **dm_pointer** entry for each secondary DM in your system. Verify that any additional **dm_pointer** entries include the correct host name.

All BRM applications now have multischema capability.

Enforcing the Creation of Unique Account Numbers

You can configure BRM to enforce the creation of unique account numbers. Similar to the **/uniqueness** object, the **/unique_account_no** object maintains a list of unique account numbers across all BRM database schemas.

To enforce the creation of unique account numbers, perform the following procedure for each CM that you want to have access to all schemas in your system:

1. Open the CM configuration file (*BRM_home*/**sys/cm/pin.conf**).
2. Edit the following entry:

   ```
   -cm uniqueness_account_no 0
   ```

   where:

   - **0** does not enforce unique account numbers. This is the default.
   - **1** enforces unique account numbers.
3. Save the file.
4. Stop and restart the CM.

Setting Up Schema Qualifications

To set up a multischema system, create schema qualifications that connect each secondary database schema to the primary database schema.

> **Note**
>
> Multidatabase Manager still creates database links between the schemas, but they are used only by the **pin_multidb** utility.

To set up schema qualifications:

1. On the primary Oracle DM machine:
   a. Open the Oracle DM configuration file (*BRM_home*/**sys/dm_oracle/pin.conf**) in a text editor.

   b. For each secondary database schema in your system, add the following entry:

   ```
   - dm schema db_no schema_name
   ```

   where *db_no* is the database number of the schema, and *schema_name* is the name of the schema.

   For example, if your system has two secondary database schemas named **pin02** and **pin03**, add the following entries:

   ```
   - dm schema 0.0.0.2 pin02
   - dm schema 0.0.0.3 pin03
   ```

   c. Save and close the file.

   d. Stop and restart the primary DM.
2. On *each* secondary Oracle DM machine:

a. Open the Oracle DM configuration file (*BRM_home*/**sys/dm_oracle/pin.conf**) in a text editor.

b. For the primary database schema and each of the other secondary database schemas in your system, add the following entry:

   ```
   - dm schema db_no schema_name
   ```

   where *db_no* is the database number of the schema, and *schema_name* is the name of the schema.

   For example, if your system has schemas named **pin01** (primary), **pin02** (secondary), and **pin03** (secondary), add the following entries:

   In the schema pin02 DM configuration file:

   ```
   - dm schema 0.0.0.1 pin01
   - dm schema 0.0.0.3 pin03
   ```

   In the schema pin03 DM configuration file:

   ```
   - dm schema 0.0.0.1 pin01
   - dm schema 0.0.0.3 pin02
   ```

c. Delete or comment out any **db_link** entries to the primary database schema. For example:

   ```
   # - dm db_link MD_PRIMARY
   ```

   > ⓘ **Note**
   >
   > A secondary DM cannot access the primary database schema by using both a schema qualification and a database link. If both entries are set, the DM reports an invalid configuration error.

d. Save and close the file.

e. Stop and restart the secondary DM.

3. Using SQL*Plus, grant each database schema the privilege to insert and update tables on the other schemas:

   a. Connect to the BRM database with SQL*Plus as **sysdba**:

      ```
      sqlplus system@databaseAlias as sysdba
      Enter password: password
      ```

      where:

      - *databaseAlias* is the Oracle system database alias.
      - *password* is the Oracle system database user password.

   b. From the primary database schema, enter the following:

      ```
      SQL> GRANT INSERT ANY TABLE TO SecondarySchema;
      SQL> GRANT UPDATE ANY TABLE TO SecondarySchema;
      ```

      where *SecondarySchema* is the name of the secondary schema.

c. From each secondary database schema, enter the following:

```
SQL> GRANT INSERT ANY TABLE TO PrimarySchema;
SQL> GRANT UPDATE ANY TABLE TO PrimarySchema;
```

where *PrimarySchema* is the name of the primary schema.

Updating Views for Multischema Systems

To update the views for your multischema systems:

1. On the primary installation system, change to the *BRM_home*/**sys/dm_oracle/data** directory.
2. Log in to SQL*Plus on the primary schema as the BRM user.
3. Run following script and commit the changes:

   ```
   SQL>@create_boc_unified_views.source
   SQL>COMMIT;
   ```

4. Log out of SQL*Plus.

What's Next?

Your BRM system now has multischema capability. You can install the BRM client applications and start creating your accounts. See "Installing BRM Thick Clients".

For information on how to manage your multischema system, such as setting database schema priority and status, see "Managing a Multischema System" in *BRM System Administrator's Guide*.

8

Installing BRM Thick Clients

Learn how to install applications that provide graphical user interfaces to the data in the Oracle Communications Billing and Revenue Management (BRM) database.

Topics in this document:

- About Installing BRM Clients
- About the BRM Client Package
- Downloading the BRM Client Software
- About Installing Self-Care Manager
- Installing BRM Clients in GUI Mode
- Installing BRM Clients in Silent Mode
- Configuring the Localization SDK on Windows
- Granting Administrative Privileges to Pricing Center Users
- Starting a BRM Client Application on Linux
- Starting a BRM Client Application on Windows
- Problems Installing BRM Client Applications
- What's Next?

> **Note**
>
> - Before installing BRM thick clients, you must first install BRM. See "Installing BRM".
> - Localized versions of BRM thick clients are supported by BRM. The following languages are supported: French, Italian, Spanish, Japanese, Korean, Chinese Simplified, Chinese Traditional, Russian, and Portuguese Brazilian.

About Installing BRM Clients

You can install the BRM clients in GUI mode or in silent mode. Installing the BRM clients in silent mode lets you perform a noninteractive installation of BRM. You can use silent mode to install the clients quickly.

BRM clients installation must be performed by a user who has permissions to write to the **oraInventory** directory.

For installation instructions, see the following sections:

- Installing BRM Clients in GUI Mode
- Installing BRM Clients in Silent Mode

About the BRM Client Package

The BRM Install Package includes the BRM server and BRM client packages. The BRM client package includes all the BRM client applications. You can use the Installer to install one or more of the following individual BRM client applications:

- Revenue Assurance Center
- Developer Center
- Customer Center (including GSM Manager: Customer Center Extension)

  GSM Manager: Customer Center Extension enables support for assigning SIM cards and telephone numbers in Customer Center. See the discussion about setting up GSM services in Customer Center Help.

  > **Note**
  >
  > You must install the GSM Manager Customer Center Extension on a system on which Customer Center is installed.

- Customer Center SDK

  You can use JBuilder (for example, Borland JBuilder) for various tasks such as creating WAR files for Self-Care Manager and custom pages for Customer Center.

- IP Address Administration Center
- Number Administration Center
- Permissioning Center
- Payment Center
- Pricing Center

  > **Note**
  >
  > When you install Pricing Center, you automatically install the Resource Editor and the Zone Mapper.

- Self-Care Manager
- SIM Administration Center
- Voucher Administration Center
- Suspense Management Center
- Localization (L10N) SDK

Downloading the BRM Client Software

You can download the BRM software from one of the following locations:

- For BRM 15.2.0 or any 15.2.*x* maintenance release: From the Oracle software delivery website (https://edelivery.oracle.com)

- For any 15.2.*x*.*y* patch set release: From the Oracle support website (https://support.oracle.com)

Search for and download the **Oracle Communications Billing and Revenue Management 15.2.*x*.*y*.0** software, where *x* refers to the maintenance release and *y* refers to the patch set release of BRM that you are installing.

The Zip archive includes one of these JAR files for installing BRM clients:

- **brmclients_15.2.*x*.*y*.0_generic.jar** contains the client software in English.
- **brmclients_all_15.2.*x*.*y*.0_generic.jar** contains the localized version of the client software.

About Installing Self-Care Manager

To use Self-Care Manager, you must install a third-party application server before installing Self-Care Manager. Supported application servers include Tomcat and WebLogic. For more information, see the vendor installation instructions for the application server you are using.

> ⓘ **Note**
>
> You must install the application server and Self-Care Manager on the same system. The Connection Manager (CM) can be on the same system as Self-Care Manager or on a different system.

For instructions on installing Self-Care Manager, see "Installing Individual BRM Clients". After installing Self-Care Manager, configure the application server to work with Self-Care Manager.

Installing BRM Clients in GUI Mode

The steps for installing BRM clients in GUI mode depend on the clients you choose to install:

- To install *all* BRM clients, select the **Complete** installation option.

  See "Installing All BRM Clients" for more information.
- To choose one or more BRM clients to install each time you run the installer, select the **Custom** installation option.

  This option installs only the clients that you select. This option is recommended for advanced users.

  See "Installing Individual BRM Clients" for more information.

Installing All BRM Clients

You can install Developer Center and Self-Care Manager on Linux, and the other BRM clients on Windows.

To install all BRM clients:

1. Obtain the BRM clients software. See "Downloading the BRM Client Software".
2. Go to the *temp_dir* directory, do one of the following:

   On Linux, run one of the following commands:

- To start the GUI installer:

  ```
  Java_home/bin/java -jar jarFile
  ```

  where:

  - *Java_home* is the directory in which you installed the latest compatible Java version.
  - *jarFile* is the BRM installer file. For example:

    ```
    brmclients_15.2.0.0.0_generic.jar
    ```

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

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

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

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

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

  where *path* is the response file location and name.

On Windows, run one of the following commands:

- To start the GUI installer:

  ```
  Java_home/bin/java -jar jarFile
  ```

  where:

  - *Java_home* is the directory in which you installed the latest compatible Java version.
  - *jarFile* is the BRM installer file. For example:

    ```
    brmclients_15.2.0.0.0_generic.jar
    ```

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

  ```
  Java_home/bin/java jarFile FilePath/oraInst.loc
  ```

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

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

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

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

The Welcome window appears.

3. Click **Next**.

   The Installation Location window appears.

4. Enter the full path or browse to the directory in which to install BRM.
5. Click **Next**.

   If the Language Selection window appears, ensure that the default language (English) is selected and click **Next**.

> ⓘ **Note**
>
> If the "Select at least one language" error appears, click **OK** and then proceed to next step.

The Installation Type window appears.

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

   The SSL Authentication window appears.

7. Do one of the following:
   - If you do not want to enable secure communication for BRM clients, deselect the **Enable SSL** check box.
   - To enable secure communication for BRM clients, leave the **Enable SSL** check box selected.
8. In the **Oracle Client Wallet Password** field, enter a password for the Oracle client wallet.
9. In the **Confirm Wallet Password** field, enter the Oracle client wallet password again.
10. Enter the client wallet password again.
11. Click **Next**.

    The Installation Summary window appears.

12. Review your selections, and click **Install**.

    The Installation Progress window appears, and the installation begins.

    > ⓘ **Note**
    >
    > After the installation begins, you cannot stop or cancel the installation.

    When the installation is done, click **Next**, the Installation Complete window 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.

    See "Troubleshooting the BRM Installation" for information about BRM installer logs.

Installing Individual BRM Clients

You can install a subset of BRM client applications (for example, Developer Center and Revenue Assurance Center) on Linux or Windows.

> ⓘ **Note**
>
> If you already have a client application installed, you must uninstall it before installing the new version.

To install individual BRM clients:

1. Obtain the BRM clients software. See "Downloading the BRM Client Software".

2. Go to the *temp_dir* directory, do one of the following:

   On Linux, run one of the following commands:

   - To start the GUI installer:

     ```
     Java_home/bin/java -jar jarFile
     ```

     where:

     - *Java_home* is the directory in which you installed the latest compatible Java version.
     - *jarFile* is the BRM installer file. For example:

       brmclients_15.2.0.0.0_generic.jar

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

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

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

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

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

     where *path* is the response file location and name.

   On Windows, run one of the following commands:

   - To start the GUI installer:

     ```
     Java_home/bin/java -jar jarFile
     ```

     where:

     - *Java_home* is the directory in which you installed the latest compatible Java version.
     - *jarFile* is the BRM installer file. For example:

       brmclients_15.2.0.0.0_generic.jar

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

     ```
     Java_home/bin/java jarFile FilePath/oraInst.loc
     ```

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

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

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

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

   The Welcome window appears.

3. Click **Next**.

   The Installation Location window appears.

4. Enter the full path or browse to the directory in which to install the BRM clients.

> **Note**
>
> If you already have a client application installed and are installing a new client application, enter the full path or browse to the directory in which the existing client application is installed.

5. Click **Next**.

   If the Language Selection window appears, ensure that the default language (English) is selected and click **Next**.

   > **Note**
   >
   > If the "Select at least one language" error appears, click **OK** and then proceed to next step.

   The Installation Type window appears.

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

   The Feature Sets Selection window appears.

7. From the clients list, select the clients to install.
8. Click **Next**.

   The SSL Authentication window appears.

9. Do one of the following:
   - If you do not want to enable secure communication for BRM clients, deselect the **Enable SSL** check box.
   - To enable secure communication for BRM clients, leave the **Enable SSL** check box selected.
10. In the **Client Wallet Password** field, enter a password for the client Oracle wallet.
11. Click **Next**.

    The Installation Summary window appears.

12. Review your selections, and click **Install**.

    The Installation Progress window appears, and the installation begins.

    > **Note**
    >
    > After the installation begins, you cannot stop or cancel the installation.

    When the installation is done, click **Next**, the Installation Complete window 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.

    See "Troubleshooting the BRM Installation" for information about BRM installer logs.

Installing BRM Clients in Silent Mode

The silent installation uses a response file in which you have set installation information. To obtain the response file, you run the GUI installer for the first install. The GUI installer generates a response file that contains the key-value pairs based on the values that you specify during the GUI installation. You can then copy and edit the response file to create additional response files for installing the BRM client applications on different machines.

Creating a Response File

To create a response file:

1. Create the response file by doing one of the following:
   - Create a copy of the response file that was generated during the GUI installation. See "Installing BRM Clients in GUI Mode" for more information.

     > **Note**
     >
     > The GUI Installer does not save passwords provided during installation in the response file. You must manually add the passwords after creating a copy of the response file.

   - Go to the *temp_dir* directory in which you downloaded the BRM client software pack, and create a response file by doing one of the following:
     – On Linux, run the following command:

       ```
       Java_home/bin/java -jar jarFile -record -destinationFile path/
       SilentInstall.rsp
       ```

       where:

       * *Java_home* is the directory in which you installed the latest compatible Java version.
       * *jarFile* is the BRM installer file. For example:

         ```
         brmclients_15.2.0.0.0_generic.jar
         ```

     – On Windows, run the following command:

       ```
       java -jar jarFile -getResponseFileTemplates -destinationFile path/
       SilentInstall.rsp
       ```

       where *jarFile* is the BRM installer file. For example:

       ```
       brmclients_15.2.0.0.0_generic.jar
       ```

   A response file is created with the default values. You can create as many response files as needed.

2. Open the file in a text editor.

3. Modify the response file you copied by specifying the key-value information for the parameters you want in your installation. In addition to any other parameters you wish to change, make sure you change/validate the following parameters:

   ```
   ORACLE_HOME
   CLIENT_WALLET_PASSWD
   CONFIRM_CLIENT_WALLET_PASSWD
   ```

4. Save and close the response file.

Performing a Silent Installation

To perform a silent installation:

1. Create a response file. See "Creating a Response File".
2. Copy the response file you created to the machine on which you will run the silent installation.
3. On the machine on which you will run the silent installation, go to the *temp_dir* directory in which you downloaded the BRM client software pack, and do one of the following:
   - On Linux, run the following command:

     ```
     Java_home/bin/java -jar jarFile -debug -invPtrLoc Inventory_home/oraInventory/
     oraInst.loc [parameter=value] -responseFile path -silent
     ```

     where:

     - *Java_home* is the directory in which you installed the latest supported Java version.
     - *jarFile* is the BRM installer file. For example:

       ```
       brmclients_15.2.0.0.0_generic.jar
       ```

     - *Inventory_home* is the location of the Oracle inventory.
     - *path* is the location and name of your response file.
     - *parameter* is the name of an installation parameter.
     - *value* is the value of the installation parameter.

     For example:

     ```
     Java_home/bin/java -jar brmclients_15.2.0.0.0_generic.jar -debug -invPtrLoc Ora/
     oraInventory/oraInst.loc INSTALL_TYPE=Complete -responseFile /tmp/
     brm_complete.rsp -responseFile path -silent
     ```

     The installation runs silently in the background.

   - On Windows, run the following command:

     ```
     java -jar jarFile -debug -invPtrLoc Inventory_home/oraInventory/oraInst.loc
     [parameter=value] -responseFile path -silent
     ```

     where:

     - *jarFile* is the BRM installer file. For example:

       ```
       brmclients_15.2.0.0.0_generic.jar
       ```

     - *Inventory_home* is the location of the Oracle inventory.
     - *parameter* is the name of an installation parameter.

- *value* is the value of the installation parameter.
- *path* is the location and name of your response file.

For example:

```
java -jar brmclients_15.2.0.0.0_generic.jar -debug -invPtrLoc Ora/oraInventory/
oraInst.loc INSTALL_TYPE=Complete -responseFile /BRM_clients/SilentInstall.rsp  -
silent
```

The installation runs silently in the background.

The Installer checks for all required software and writes errors to a log file if it detects any missing or unavailable components or if any connectivity issues occur.

For information about BRM installer logs, see "Troubleshooting the BRM Installation".

Configuring the Localization SDK on Windows

After installing the localization (L10N) SDK, you must configure the SDK on Windows. See "Installing Individual BRM Clients" for installing the Localization SDK.

To configure the localization SDK on Windows, in a command window, use the **subst** command to substitute the drive letter for the build tree directory of the SDK.

If you installed the SDK in the default location, enter the command as follows:

```
subst W: "C:\Oracle_Home\LocalizationSDK"
```

Granting Administrative Privileges to Pricing Center Users

You use the **pricing_admin.pl** script to grant administrative privileges to Pricing Center users. Users with administrative privileges are called pricing admins.

To grant administrative privileges to Pricing Center users, perform the procedures in the following sections:

1. Modifying the Database Configuration File
2. Modifying the pricing_admin.pl Script Configuration File
3. Setting Up the Database Server for the Pipeline Manager Administrator
4. Initializing the Pricing Admin Configuration Object
5. Specifying Administrative Privileges for Pipeline Manager Users
6. Removing Pricing Admins

Modifying the Database Configuration File

1. Go to the DM directory:

   ```
   cd BRM_home/sys/dm_oracle
   ```

2. Back up the database configuration file (**pin.conf**).

   > **Note**
   >
   > You will be restoring the **pin.conf** file later in the procedure.

3. Open the **pin.conf** file with a text editor.
4. If the **dd_write_enable_objects** entry is set to **0**, set the entry to **1**.
5. If the **crypt** entry is commented out, uncomment it and configure it according to the notes that precede this entry:

   ```
   - crypt Encrypt_method| BRM_home/lib/Encryption_library "Secret_key"
   ```

   where:

   - *Encrypt_method* is the type of encryption method, which is either **aes** or **ozt**.
   - *Encryption_library* is the path and filename of the encryption library. The prefix for the library is **lib** for Linux, or **null ""** for Windows. The extension for the library is **.so** for Linux, and **.dll** for Windows.
   - *Secret_key* is your encrypted AES or OZT key.

   For more information, see "Configuring the Data Manager for AES Encryption" or "Configuring the Data Manager for Oracle ZT PKI Encryption" in *BRM Developer's Guide*.
6. Save and close the file.
7. Stop and restart the DM and CM. See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

Modifying the pricing_admin.pl Script Configuration File

To modify the **pricing_admin.pl** script's configuration file:

1. Go to the *BRM_home***\setup\scripts** directory.
2. Open the setup configuration file (**pin.conf**) file with a text editor.
3. Check that the following entries are specified in the **pin.conf** file:

   ```
   - nap login_type 1
   - - userid 0.0.0.1 /service/admin_client 1
   ```
4. Specify appropriate values for **login_name** and **login_pw**:

   ```
   - nap login_name login_name
   - nap login_pw password
   ```

   For example:

   ```
   - nap login_name integrate
   - nap login_pw integrate
   ```
5. Enter the host name and port number of your server in the **cm_ptr** entry:

   ```
   - nap cm_ptr ip host_name port_number
   ```
6. If you are using Pipeline Manager, specify values for the entries described in Table 8-1. These entries specify Pipeline Manager configuration data used by the **pricing_admin.pl** script to set up and maintain pricing admins and the Pipeline Manager database for Pricing Center.

Table 8-1 Pipeline Manager Configuration Data

Entry	Description
default_table_space_name	The tablespace that newly created Oracle users are assigned to. The default is **INTEGRATE_TS_1_DAT**. If the default tablespace is not available in the Pipeline Manager database, use an available existing tablespace or create a new one. **Important:** You must specify a value for **default_table_space_name** before you run the **pricing_admin.pl** script with the **-set** parameter.
database_role_name	The role that provides select, insert, delete, and update privileges on IFW_* and JSA_* tables. The default is **INTEGRATE_ROLE_ALL**. If the default role is not available in the Pipeline Manager database, use an existing role that provides these privileges or create a new role. If you use the **-init** parameter, the entry is optional.
host	The host name for the server where the Pipeline Manager database is located and running.
port	The port number used by the Pipeline Manager database listener.
db	The SID for the Pipeline Manager database instance. **Note:** You must set the **db** value to the ORACLE_SID value. If you set **db** to the SERVICE_NAME value, BRM does not generate an error when creating the **/config/pricing_admin** object; however, you cannot connect to the Pipeline Manager database from Pricing Center.
db_type	The name of the database vendor. The value should be one of the system database driver groups of entries in the **jsaconf.properties** file, for example, **oracle**.
login_name	The default Pipeline Manager database user name. **Important:** You must specify a value for **login_name** before you run the **pricing_admin.pl** script with the **-init** parameter.
login_pw	The password of the default Pipeline Manager database user. **Important:** You must specify a value for **login_pw** before you run the **pricing_admin.pl** script with the **-init** parameter.
admin	The DBA on the Pipeline Manager database that has user creation privileges.
admin_pw	The password of the DBA account.
db_alias	The net service name for the Pipeline Manager database in your **tnsnames.ora** file.

The following shows sample **pin.conf** entries:

```
- pipeline default_table_space_name     INTEGRATE_TS_1_DAT
- pipeline database_role_name           INTEGRATE_ROLE_ALL
- pipeline host                         ttal-013
- pipeline port                         1521
- pipeline db                           mySID
- pipeline db_type                      oracle
- pipeline login_name                   integrate
- pipeline login_pw                     integrate
- pipeline admin                        system
- pipeline admin_pw                     manager
- pipeline db_alias                     pindb
```

> **Note**
>
> To ensure that Pricing Center can connect to the server, add the name and IP address of the Pipeline Manager host system to the host file of each machine running Pricing Center.

7. Save and close the file.

Setting Up the Database Server for the Pipeline Manager Administrator

To set up an Oracle database server for the Pipeline Manager administrator:

1. Edit the **tnsnames.ora** file to specify HOST, PORT, and SERVICE_NAME values. Use the same values that you used for the **host**, **port**, and **db** entries in the setup configuration file (*BRM_home***setup\scripts\pin.conf**).

   The following is the required name entry in the **tnsnames.ora** file that corresponds to the sample **pin.conf** file shown in "Modifying the pricing_admin.pl Script Configuration File".

   > **Note**
   >
   > The database SID in the following example is **mySID**, the port name is **1521**, and the net service name for the Pipeline Manager database is **pindb**. Be sure to change these values to match your environment.

   ```
   mySID =
    (DESCRIPTION =
     (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = ttal_013) (PORT = 1521) )
     )
    (CONNECT_DATA =
     (SERVICE_NAME = pindb)
    )
   )
   ```

2. To ensure that the **pin.conf** entries for **admin**, **admin_pw**, **db_alias**, and **host** are correct, try connecting to the Pipeline Manager database using the following command:

   ```
   sqlplus admin/admin_pw@db_alias
   ```

Initializing the Pricing Admin Configuration Object

Initialize the pricing admin configuration object by running the **pricing_admin.pl** script:

1. Go to the *BRM_home***setup\scripts** directory.
2. Run the **pricing_admin.pl** script with the **-init** parameter:

   ```
   perl pricing_admin.pl -init
   ```

   If initialization is successful, the system returns this line:

   ```
   Initialized config object for pricing admin.
   ```

3. Restore the original database configuration file (**pin.conf**) that you backed up in "Modifying the Database Configuration File".

4. Stop and restart the DM to apply the new settings. See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

Specifying Administrative Privileges for Pipeline Manager Users

To specify administrative privileges for Pipeline Manager users:

1. Ensure that every Pipeline Manager user who will be a pricing admin has a CSR account in BRM (an account using the **admin_client** service). You create CSR accounts in Customer Center by using the CSR Plan. See information about the Account Creation wizard's Contact page in Customer Center Help.
2. Create a text file that has the user name and password (in that order) of each pricing admin on a separate line. User names and passwords must be separated by a space. For example:

```
john_jones password
mary_allen password
```

> **Note**
>
> BRM user names (login names) are case sensitive, but Pipeline Manager user names are not. Therefore, type the names exactly as specified in Customer Center, but avoid using names that are too similar. For example, do not create two accounts with the user names ”Smith” and ”smith” because they will both be logged in to Pipeline Manager as ”SMITH.”

3. Go to the *BRM_home*\setup\scripts directory.
4. Run the **pricing_admin.pl** script with the **-set** parameter:

```
perl pricing_admin.pl -set < text_file
```

   where *text_file* is the name of the file you created in Step 2.

5. Delete the text file if security is a concern.

Removing Pricing Admins

This section describes how to remove single and multiple users from the Pipeline Manager database.

Removing One Pricing Admin

To remove one pricing admin from the database:

1. Go to the *BRM_home*\setup\scripts directory.
2. Run the **pricing_admin.pl** script with the **-remove** parameter:

```
perl pricing_admin.pl -remove
```

3. When prompted, enter the user name to be deleted.

   For each pricing admin removed, the system responds with this line:

```
Removed pricing admin user_name
```

4. Repeat Step 3 until all appropriate user names are removed.

Removing Multiple Pricing Admins

To remove multiple pricing admins from the database:

1. Go to the *BRM_home***\setup\scripts** directory.
2. Create a text file (*text_file*) containing only the user names that you want to remove. Put each user name on a separate line, as in this example:

   ```
   john_jones
   mary_allen
   ```

3. Run the **pricing_admin.pl** script with the **-remove** and *text_file* parameters:

   ```
   perl pricing_admin.pl -remove < text_file
   ```

Starting a BRM Client Application on Linux

You can install Developer Center and Self-Care Manager on Linux only.

To start a BRM client application on Linux:

1. Go to the *install_dir* directory, where *install_dir* is the directory in which you installed the BRM clients.
2. Grant appropriate read and write permissions for the BRM client application that you want to start.
3. Run the following command:

   ```
   ./applicationName.sh
   ```

   where *applicationName* is the name of the BRM client application you want to start.

Starting a BRM Client Application on Windows

To start a BRM client application on Windows:

1. From the start options, choose *ProductName*, where *ProductName* is the name of the BRM client application you want to start.

   The Login dialog box appears.

2. Enter your login name and password.

   For information about login names and passwords for client applications, see "Implementing System Security" in *BRM System Administrator's Guide*.

3. To connect to a database other than the one specified during installation, click **Connection Info** and enter the host name and port number.
4. Click **OK**.

Problems Installing BRM Client Applications

This section describes problems you might encounter when installing BRM client applications and their solutions.

Problem: Cannot Start a Client Application after Installation

You cannot start one of the client applications, even though the installation appeared to be successful.

Possible Cause

BRM could not add the client application to your Windows path because the path is too long. Therefore, Windows cannot find it.

Solution

Review your path statement and delete references to obsolete programs. For more information, see your Windows documentation. Reboot the system before you try restarting the client application.

What's Next?

After installing the BRM client applications, you can test BRM with your business models:

- For an overview of setting up product offerings, see "About Creating Product Offerings" in *BRM Creating Product Offerings*.
- For an overview of setting up billing, see "Configuring Billing" in *BRM Configuring and Running Billing*.
- For an overview of setting up customer account creation and managing customer accounts, see "Customizing Account Creation" in *BRM Managing Customers*.

9

Installing Pipeline Manager

Learn how to install the Oracle Communications Billing and Revenue Management (BRM) Pipeline Manager.

Pipeline Manager Installation Overview

To install and configure Pipeline Manager, follow the steps in these sections:

1. Pipeline Manager System Requirements
2. Pipeline Manager Preinstallation Tasks
3. Installing Pipeline Manager
4. Increasing Heap Size to Avoid “Out of Memory" Error Messages
5. Configuring an Oracle Pipeline Manager Database

Pipeline Manager System Requirements

Before you install and run Pipeline Manager, ensure that you have installed and configured the required hardware and software.

Pipeline Manager requires the following software:

- Oracle Database server
- Oracle Database client
- Java Runtime Environment (JRE)

See *BRM Compatibility Matrix* for the supported software versions.

The number and configuration of the computers that you employ for your Pipeline Manager installation depend on the scale and the kind of deployment you have planned.

> **Note**
>
> The more pipelines you configure and the more CPUs you use, the more important it is to put input and output on systems that are managed by different Controllers.

Pipeline Manager Preinstallation Tasks

Perform the following tasks before installing Pipeline Manager:

1. Install and configure BRM. See "Installing and Configuring BRM" for more information.
2. Install the Pipeline Manager database. See "Configuring an Oracle Pipeline Manager Database".

3. Configure the OS environment. See "Creating a User and Configuring Environment Variables" for more information.
4. Update the OS kernel. See "Setting the Maximum Allowed Number of Open Files" for more information.

Creating a User and Configuring Environment Variables

Follow these steps to create a Pipeline Manager user and set up system environment variables on a system:

1. Create a user that owns the Pipeline Manager software. The examples in this document use the default user **integrate** with the bash shell as the default shell.

   > **Note**
   >
   > You must create a Pipeline Manager user and set the user's environment before installing Pipeline Manager.

2. Log in as user **integrate**.
3. Go to the *Pipeline_home* directory.
4. Open the **source.me.sh** file for your shell in a text editor.

   > **Note**
   >
   > The **source.me** file is for a bash shell. If you use a C shell, open the **source.me.csh** file.

5. Set the environment variables for Pipeline Manager that are listed in Table 9-1:

Table 9-1 Pipeline Manager Environment Variables

Environment Variable	Description
IFW_HOME	The directory where the Pipeline Manager software is installed. **Note:** This documentation refers to this directory as *Pipeline_home*.

LD_LIBRARY_PATH	The library path.
	Set this to include *Pipeline_home*/**lib** at the end of the variable. For example, in bash set the following:
	`export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:Pipeline_home/lib`
	Important: After your first upgrade reset the LD_LIBRARY_PATH environment variable to the following:
	`setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH};BRM_home/lib:Pipeline_home/lib`
	where *BRM_home*/**lib** is before *Pipeline_home*/**lib**.

Table 9-1 (Cont.) Pipeline Manager Environment Variables

Environment Variable	Description
MALLOC_TRIM_THRESHOLD_ **MALLOC_MMAP_MAX_** **MALLOC_TOP_PAD_**	On Linux Pipeline Manager platforms, modify the MALLOC_TRIM_THRESHOLD_, MALLOC_MMAP_MAX_ and MALLOC_TOP_PAD_ environment variables with these values: • MALLOC_TRIM_THRESHOLD_ = **-1** • MALLOC_MMAP_MAX_ = **0** • MALLOC_TOP_PAD_ = **268435456**
MAX28DIGIT_DECIMAL_SUPPORT	By default, Pipeline Manager supports 15-digit decimal. To configure Pipeline Manager to support up to 28-digit decimal, set this to **Y**.
NLS_LANG	Set this to **LANG American_America.AL32UTF8**. **Important**: You must use **American_America** as the language and territory regardless of your locale and the **UTF8 or AL32UTF8** character set. BRM supports AL32UTF8 as its default character set. It also continues to support the UTF8 character set for BRM installations that are being upgraded from previous versions. The unicode character set AL32UTF8 is recommended for all new BRM deployments.
PATH	The Pipeline Manager executable path. Set this to include *Pipeline_home***/bin**.
IFW_EVENTHANDLER_PORT	The port number on which the event handler daemon listens for events. **Important:** • If you are starting more than one framework process, change the value of this parameter before starting each process. Each process must have a unique event handler port for all users on a host. A framework process will not start if it cannot start the event handler daemon, and the event handler daemon will not start if it cannot listen on the specified port. • Set this variable to a port number that is not in use. Do not set it to a well-known port number, such as port 11960 which is used by BRM.

6. Save and close the updated file.
7. Update the environment for the current shell session:

   For bash shell:

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

   C shell:

   ```
   source source.me.csh
   ```

Setting the Maximum Allowed Number of Open Files

To avoid causing a failure of Pipeline Manager, you must configure the maximum number of pipeline files allowed in the kernel.

To set the maximum open files on Linux:

1. Log on as **root**.
2. Open the system file (**/etc/sysctl.conf**).
3. Add the following lines to the end of the file, or modify the values if these lines are already present. For example:

   ```
   # sets the hard limit on file descriptors:
   fs.file-max = 2605192
   ```

4. Run **sysctl -p** to reload all the settings from the system file.

Configuring an Oracle Pipeline Manager Database

This section describes how to configure the Pipeline Manager database on Oracle.

Before proceeding, you should be familiar with executing database scripts.

Before setting up the Oracle database for Pipeline Manager, ensure the following:

- An Oracle database instance is mounted and open.
- You have administrator access to the database instance.

To install and configure the Pipeline Manager database, follow the steps in these sections:

1. Setting the Environment for the Pipeline Manager Database
2. Setting Up the Oracle JSA Database Schema
3. Setting Up the Oracle Pipeline Manager Framework Database Schema
4. Changing Public Synonyms to Private for Users in Multiuser Environments
5. Loading Procedures for FCT_DuplicateCheck

Setting the Environment for the Pipeline Manager Database

Before setting up the database schemas for your Oracle Pipeline Manager database, you must configure database environment variables.

1. Open the configuration file for the login shell. This is the file in which your user profile is stored; for example, **.profile** or **.bashrc**. By default it is stored in your home directory.
2. Change the variables in Table 9-2 to match your Pipeline Manager environment:

Table 9-2 Environment Variables of Pipeline Manager Database

Environment Variable	Description
ORACLE_HOME	Set this variable to point to the Oracle database installation path.
LD_LIBRARY_PATH	Set this to point to the database **/lib** directories, for example: `LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib:$ORACLE_HOME/rdbms/lib`
LD_LIBRARY_PATH_64	Set this to point to the database **/lib** directories, for example: `LD_LIBRARY_PATH_64=$IFW_HOME/lib:$ORACLE_HOME/lib:$ORACLE_HOME/rdbms/lib`
PATH	Set this to the database **/bin** directory, for example: `PATH=$PATH:$ORACLE_HOME/bin`

3. Save and close the file.

Setting Up the Oracle JSA Database Schema

This section describes how to run scripts to set up the Oracle JSA database schema.

Run the scripts described in the following steps to set up the database schema for the JSA tables and tablespaces:

> **Note**
>
> All scripts are located in the *Pipeline_home***/database/Oracle/Scripts** directory.

1. Open the **JSA_Tablespaces.sql** file with a text editor.
2. Replace all occurrences of the string *** with the path to the Oracle tablespace data file.
3. Save the file.
4. Log in as user **database administration (DBA)**.
5. Run the **JSA_Tablespaces.sql** script.

   This script creates the tablespaces for the JSA server.
6. Run the **JSA_Roles.sql** script.

   This script creates roles and the default JSA user.
7. Log out and log in as user **JSA**.
8. Run the **JSA_Create.sql** script.

   This script creates the JSA tables.
9. Run the **JSA_Synonyms.sql** script.

   This script creates the JSA public synonyms.
10. Run the **JSA_Prepare.sql** script.

    This script inserts some default values into the tables.

The Oracle JSA database schema is now set up.

Setting Up the Oracle Pipeline Manager Framework Database Schema

To set up the database schema for the Pipeline Manager framework work tables and tablespaces:

> **Note**
>
> All scripts are located in the *Pipeline_home***/database/Oracle/Scripts** directory.

1. Open the **ifw_Tablespaces.sql** file with a text editor.
2. Replace the occurrences of the string **ORA_DAT_PATH** and **ORA_IDX_PATH** with the path that leads to the ***.dbf** files shown in the script file.
3. Save the file.
4. Log in as user **DBA**.
5. Run the **ifw_Tablespaces.sql** script.

   This script creates the tablespaces for the Pipeline Manager framework.
6. (Optional) Change the default Pipeline Manager user name (*pipeline_user*) from **integrate** to another name:

   a. Open the **ifw_Roles.sql** file by using a text editor.

b. In the User: INTEGRATE section, replace the two occurrences of INTEGRATE in the following string with the new default user name:

```
INTEGRATE identified by INTEGRATE
```

c. In the GRANT commands that follows the QUOTA commands in the User: INTEGRATE section, replace all occurrences of INTEGRATE with the new default user name.

d. Save the file.

7. Run the **ifw_Roles.sql** script.

   This script creates the appropriate roles and the default user.

8. Log out and log in as *pipeline_user*.

9. Run the **ifw_Create.sql** script.

   This script creates the Pipeline Manager framework tables.

10. Run the **ifw_Synonyms.sql** script.

    This script creates the public synonyms.

The Oracle Pipeline Manager framework database schema is now set up.

You can create only one instance of the pipeline schema in a particular database. You can create multiple users and give permission to access the same tables. Creating multiple instances lead to public synonym conflict.

To avoid public synonym conflicts in a multiuser environment, change the public synonyms to private synonyms for specific users. See "Changing Public Synonyms to Private for Users in Multiuser Environments" for more information.

For example, if you create **table1** for **user1**, you can create **user2** and assign permissions to access **table1**. Typically, you access the table as **user1.table1**. However, using synonyms, you can directly access the table as table1. No prefix (**user1**) is required.

Changing Public Synonyms to Private for Users in Multiuser Environments

Changing public synonyms to private synonyms for a user is required for avoiding public synonym conflict in a multiuser environment.

> **Note**
>
> All scripts are located in the *Pipeline_home*/**database**/**Oracle**/**Scripts** directory

To change public synonyms to private synonyms for a user:

1. Connect to the Oracle database with SQL*Plus:

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

   where *databaseAlias* is the database alias of the Oracle database.

2. Retrieve the list of public synonyms to drop by running the following command:

   ```
   select synonym_name from all_synonyms where
         owner = 'PUBLIC' and table_name like '%IFW%';
   ```

3. Drop the public synonyms returned in the previous step for the user using commands like the following:

   ```
   drop public synonym IC_DAILY;
   drop public synonym IC_DAILY_ALTERNATE;
   drop public synonym IFW_AGGREGATION;
   . . .
   ```

4. Grant CREATE ANY synonym access to the user you created for the Pipeline Manager tables. For example:

   ```
   SQL> CREATE ROLE synonym_role;
   SQL> GRANT CREATE ANY SYNONYM TO synonym_role;
   SQL> GRANT synonym_role TO pin_user;
   ```

   where *pin_user* is the BRM user.

5. Open the **ifw_Synonyms.sql** file with a text editor.
6. Remove the occurrences of the string public and add the Pipeline Manager database user. For example, for ifw_SEQ_AGGREGATION table, change the string to:

   ```
   create synonym ifw_SEQ_AGGREGATION for ifw_user.ifw_SEQ_AGGREGATION;
   ```

   where *ifw_user* is the is the Pipeline Manager database user.

7. Save the file.
8. Run the **ifw_Synonyms.sql** script.

   This script creates the private synonyms for the user.

9. Exit SQL*Plus.

The public synonyms are now changed to private synonyms for the user.

Loading Procedures for FCT_DuplicateCheck

Before using the FCT_DuplicateCheck module, you must load the duplicate check stored procedures in the Pipeline Manager database:

1. Load the **DuplicateCheck_Oracle.plb** file from *Pipeline_home***/database/Oracle/Scripts/DuplicateCheck**.
2. Verify that there is an unique index BIDX_DUPCHK_DATA on the DATA column of the IFW_DUPLICATECHECK table. If not, then create it before starting Pipeline Manager.

The duplicate check stored procedures are now loaded.

Installing Pipeline Manager

This section describes how to install Pipeline Manager.

> ⓘ **Note**
>
> If you are installing Pipeline Manager to replace an identical release (for example, to restore a clean version of the package), you must first uninstall the existing installation. See "Uninstalling Pipeline Manager and Optional Components".

To install Pipeline Manager, see "Installing Individual BRM Components".

> **Note**
>
> Ensure that SSL/TLS is enabled in real-time and batch pipelines. See Enabling SSL/TLS in Real-Time Pipeline and Enabling SSL/TLS in Batch Pipeline in *BRM System Administrator's Guide*.

To upgrade Pipeline Manager, see "Upgrading BRM 12.0 to BRM 15.2".

Pipeline Manager Directory Structure

Installation creates the following directory structure shown in Table 9-3:

> **Note**
>
> *Pipeline_home* is the default Pipeline Manager installation directory.

Table 9-3 Directory Structure Created by Pipeline Manager Installation

Directory	Description
*Pipeline_home***/bin/**	Server binaries, such as the main Pipeline Manager binary **ifw**.
*Pipeline_home***/conf/**	Contains registry files.
*Pipeline_home***/data/**	Contains source and directory files for EDRs.
*Pipeline_home***/database/**	Database creation scripts and database documentation.
*Pipeline_home***/discount/**	Balance Data Module transaction and data.
*Pipeline_home***/etc/**	Error messages. **Note:** You can copy the error messages to another directory and customize them; for example, as preparation for translation into other languages.
*Pipeline_home***/formatDesc/**	External file format definitions and internal container definitions.
*Pipeline_home***/info/**	Info registry, which shows the current system status.
*Pipeline_home***/instrumentation/**	Performance Data collection.
*Pipeline_home***/iScriptLib/**	iScript libraries.
*Pipeline_home***/lib/**	Shared libraries.
*Pipeline_home***/lib64/**	Shared libraries (64-bit)
*Pipeline_home***/log/**	Process log.
*Pipeline_home***/releaseNotes/**	Release notes.
*Pipeline_home***/samples/**	Sample registries that you can use for testing your Pipeline Manager setup.
*Pipeline_home***/semaphore/**	Directory that is checked regularly for semaphores.
*Pipeline_home***/tools/**	The tools such as binaries and Perl scripts.

Installing Pipeline Manager Optional Components

This section describes how to install the following Pipeline Manager optional components:

- Pipeline Configuration Manager
- Pipeline PDK Manager
- Interconnect Manager
- CIBER Roaming Manager (requires Interconnect Manager)
- TAP Roaming Manager (requires Interconnect Manager)

For more information on configuring roaming in Pipeline Manager, see "Setting Up Pipeline Manager for Roaming Incollect Processing" and "Setting Up Pipeline Manager for Roaming Outcollect Processing" in *BRM Implementing Roaming*.

Before installing Pipeline Manager optional components, you must install Pipeline Manager.

> ⓘ **Note**
>
> If you are installing the optional components to replace an identical release (for example, to restore a clean version of the package), you must first uninstall the existing installation. See "Uninstalling Optional Components".

To install Pipeline Manager optional components, see "Installing Individual BRM Components".

> ⓘ **Note**
>
> If you install Pipeline PDK Manager, after installing it consult the *PDK_home***/ README.PDK** file, where *PDK_home* is the directory in which the PDK is installed, for additional steps to follow before using the PDK.

Increasing Heap Size to Avoid “Out of Memory" Error Messages

To avoid "Out of Memory" error messages in the log file after installation, increase the maximum heap size used by the Java Virtual Machine (JVM). The exact amount varies greatly with your needs and system resources. By default, the JVM used has a maximum heap size of 60 MB. Increase the maximum heap size to 120 MB by entering the following sample code in a text editor:

```
%IF_EXISTS%("INIT_JAVA_HEAP", "@INIT_JAVA_HEAP@20m") %IF_EXISTS%("MAX_JAVA_HEAP",
"@MAX_JAVA_HEAP@120m")
```

where **20m** and **120m** indicate the minimum and maximum heap sizes respectively.

Save the file as *Packagename***.ja** in the temporary directory (*temp_dir*) to which you downloaded the installation software.

Packagename indicates the name of the installation software.

Uninstalling Pipeline Manager and Optional Components

To uninstall Pipeline Manager and its optional components, see "Uninstalling Optional Components".

10 Installing Pipeline Configuration Center

Learn how to install Pipeline Configuration Center (PCC), which is a Web-based application that serves as the user interface for Pipeline Manager, on top of the Oracle Communications Billing and Revenue Management (BRM) software.

> **Note**
>
> Localized versions of PCC are supported by BRM. The following languages are supported: French, Italian, Spanish, Japanese, Korean, Chinese Simplified, Chinese Traditional, and Portuguese Brazilian.

Topics in this document:

- PCC Installation Overview
- PCC System Requirements
- PCC Preinstallation Tasks
- Installing PCC
- PCC Postinstallation Tasks
- Verifying the PCC Installation
- Uninstalling PCC
- Administering PCC
- Troubleshooting PCC
- Secure Deployment Checklist

PCC Installation Overview

PCC installation should be performed only by experienced system administrators. You must be familiar with the following before you begin the installation:

- Linux operating system
- Oracle WebLogic Server administration
- Oracle Database administration

Additionally, you should have experience installing Java-related packages.

To learn more about PCC installation, see the following:

- About the PCC Architecture
- Overview of the PCC Installation Procedure
- Performing a Secure PCC Installation
- Ensuring a Successful PCC Installation

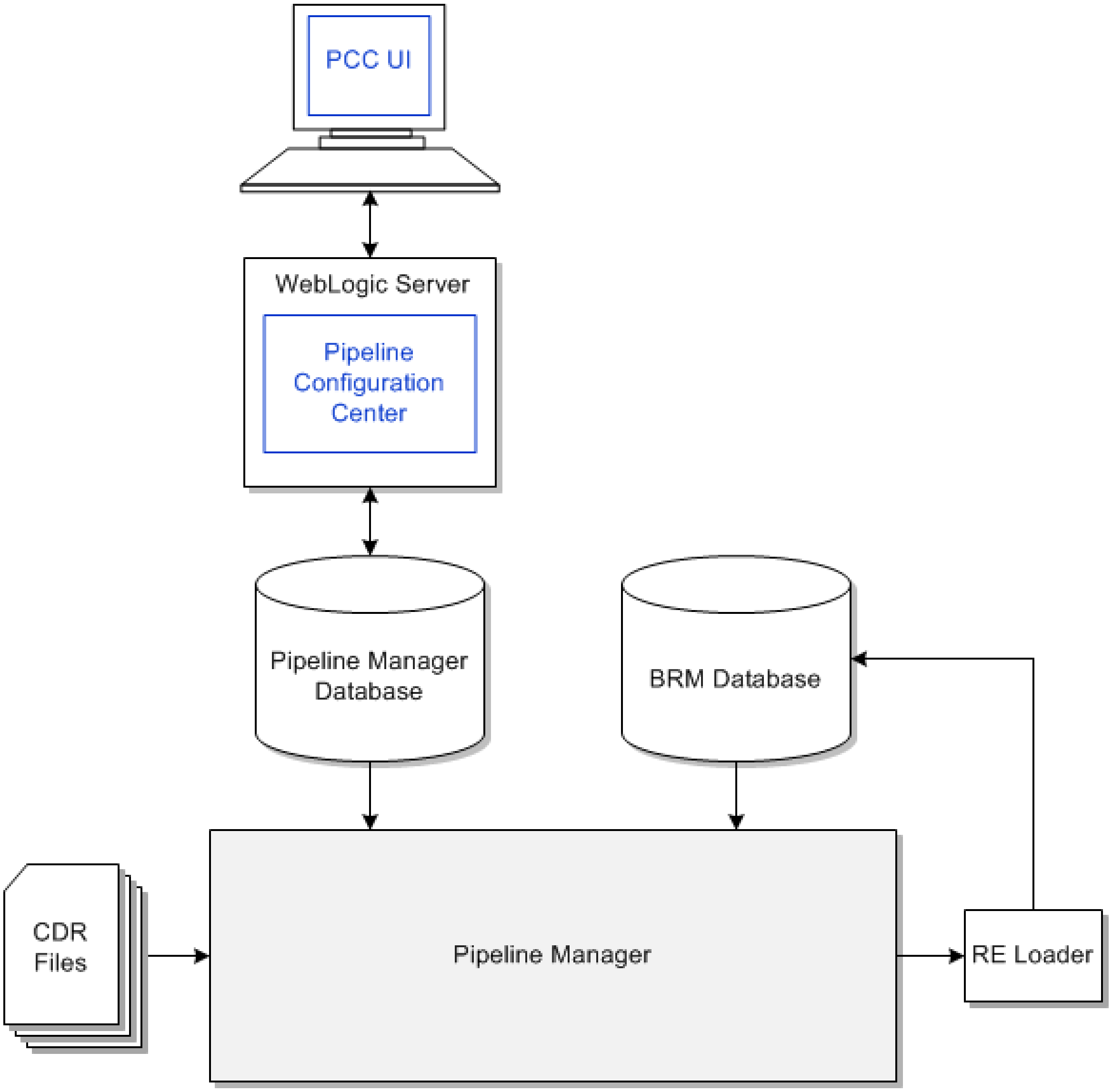
About the PCC Architecture

PCC is a Web-based BRM application that serves as the user interface for Pipeline Manager.

PCC runs on a managed server in an Oracle WebLogic Server domain. To interact with PCC, users access the PCC login page URL in a browser and then enter their PCC user name and password in the login page. After a user is authenticated, the PCC home page appears in the browser.

Figure 10-1 shows how PCC and Pipeline Manager are integrated.

Figure 10-1 PCC and Pipeline Manager Integrated Architecture



Overview of the PCC Installation Procedure

The following is an overview of the PCC installation procedure:

1. Plan your installation. Planning an installation involves the following:
   - Determining the scale of your implementation. For example, is it a small test system or a large production system?
   - Assessing how many physical computers you need and which software components to install on which computers.
2. Review system requirements.
3. Perform the following preinstallation tasks:
   - Install and configure Oracle WebLogic Server.
   - Set the JAVA_HOME environment variable.
   - Install and configure BRM.
   - Configure Oracle Database advanced security encryption and integrity algorithms for a secure connection from the installer.
4. Install PCC.
5. Perform the postinstallation configuration tasks.
6. Verify the installation.

Performing a Secure PCC Installation

You can install PCC in the following modes:

- GUI mode
- Silent mode

Oracle recommends that you install PCC using Secure Sockets Layer (SSL) in a production environment. See "Installing PCC" for more information.

The silent mode installation is not meant for production environments. It should be used only in test environments and only for setting up quickly or backing up the properties for later use in another test environment.

See "Secure Deployment Checklist" for information about the checklist to deploy PCC securely.

Ensuring a Successful PCC Installation

To ensure that the PCC installation is successful, follow these guidelines:

- As you install each component (for example, the WebLogic Server), verify that the component is installed successfully before continuing the installation process.
- Pay close attention to the system requirements.

  Before you begin installing the application, ensure that your system has the required software. In addition, ensure that you know all the required configuration values, such as host names and port numbers.
- Make a note of any new configuration values as you create them.

  You will be required to enter configuration values later.

PCC System Requirements

Before you install and run PCC, ensure that you have installed and configured the required hardware and software.

PCC requires the following software:

- Oracle Database server
- Oracle Database client
- Java Runtime Environment (JRE)
- Oracle WebLogic Server
- Oracle Fusion Middleware Application Development Framework
- Internet Browser

See *BRM Compatibility Matrix* for the supported software versions.

The number and configuration of the computers that you employ for your PCC installation depend on the scale and the kind of deployment you have planned.

PCC Preinstallation Tasks

Perform the following tasks before installing PCC:

1. Install and configure Oracle WebLogic Server. See "Installing and Configuring Oracle WebLogic Server".
   - Enable SSL for the target Oracle WebLogic Server server domain, configure the server KeyStore certificate, and get the client KeyStore trusted certificate (**.p12** [recommended] or **.jks** file).
   - If SSL is enabled, ensure that the KeyStore file is created in a secure drive and access is strictly limited to the user account.
2. Set the JAVA_HOME environment variable. See "Setting the JAVA_HOME Environment Variable".
3. Download and save the JDBC driver to your PCC system. See "Downloading the JDBC Driver for PCC".
4. Install and configure BRM. See "Installing and Configuring BRM".
5. Configure Oracle Database advanced security encryption and integrity algorithms for a secure connection from the installer. See the Oracle Database documentation for advanced security configuration parameters. This is required for the PCC installer to make a secured (encrypted) database connection over the network.
6. Download the PCC software. See "Downloading the PCC Software".

Installing and Configuring Oracle WebLogic Server

This section describes procedures related to installing Oracle WebLogic Server and configuring the Oracle WebLogic Server domain on which you will deploy PCC.

See the information in the following sections:

1. Installing Oracle WebLogic Server

2. Installing and Configuring Oracle Application Development Runtime
3. Creating an Oracle WebLogic Server Domain
4. Configuring SSL for the Oracle WebLogic Server Domain

Installing Oracle WebLogic Server

Oracle WebLogic Server is available as a component of the Oracle Fusion Middleware software. For more information on supported versions, see *BRM Compatibility Matrix*.

Download Oracle WebLogic Server from the Oracle software delivery website (https://edelivery.oracle.com).

See *Installing and Configuring Oracle WebLogic Server and Coherence* for information about installing WebLogic Server.

Installing and Configuring Oracle Application Development Runtime

Oracle Application Development Runtime must be installed on the computer on which you install PCC. You can download Oracle Application Development Runtime from the Oracle software delivery website:

`http://edelivery.oracle.com`

For information on installing Oracle Application Development Runtime, see the Oracle Application Development Runtime documentation.

> **Note**
>
> The Oracle Fusion Middleware Application Developer installer installs both Oracle Application Development Runtime and Oracle Enterprise Manager.

See *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework* for more information on Application Development Framework.

Creating an Oracle WebLogic Server Domain

To create a domain on which to deploy PCC, see the Oracle WebLogic Server documentation.

You must create the domain with at least one managed server and one administration server. Oracle recommends that you deploy PCC on a managed server in a production environment.

Table 10-1 shows the values to choose when creating the domain for PCC.

Table 10-1 Oracle WebLogic Server Domain Configuration Values

Configuration Value	Description
Domain Source	Select **Generate a domain configured automatically to support the following BEA products**. From the list, select **Oracle JRF - 14.1.2 [oracle_common]**.
Domain Mode Configuration	Select **Production Mode**.
Optional Configuration	Select **Administration Server, Managed Servers, Clusters and Machines** and **Deployments and Services**.

Table 10-1 (Cont.) Oracle WebLogic Server Domain Configuration Values

Configuration Value	Description
Administration Server Listen Address	Use a listen address that is equal to a resolvable DNS host or IP address. Do not use **localhost** or **127.0.0.1**. Those addresses interfere with clustered servers.
Managed Server Listen Address	Enter a listen address that is equal to a resolvable DNS host or IP address.
Target Deployments to Clusters or Servers	Add all libraries to both the managed server target and the Administration Server.
Target Services to Clusters or Servers	Add all services to the managed server target.

See *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Content Management Suite* for more information on Oracle WebLogic Server domains.

You can now start the domain administration server manually and log in to Oracle WebLogic Remote Console.

Configuring SSL for the Oracle WebLogic Server Domain

To enable secure communication for PCC, you must configure secure socket layer (SSL) for the domain on which you want to deploy PCC.

For more information about configuring SSL in WebLogic Server, see "Configuring SSL" in *Administering Security for Oracle WebLogic Server*.

Pipeline Configuration Center supports TLS versions 1.2 and 1.3. For information about setting the protocol version, see "Specifying the SSL/TLS Protocol Version" in *Administering Security for Oracle WebLogic Server*.

Setting the JAVA_HOME Environment Variable

You must set the JAVA_HOME environment variable to your version of Java on the PCC machine and for the WebLogic Server domain.

See "BRM Software Compatibility" in *BRM Compatibility Matrix* for the compatible version of Java.

To set the JAVA_HOME environment variable on the machine on which you want to deploy PCC:

1. On the machine on which you want to deploy PCC, set the JAVA_HOME environment variable to the directory in which the JRE/JDK is installed.
2. Verify the Java version by running the following command:

   ```
   $JAVA_HOME/bin/java -version
   ```

   The Java version is displayed.

To set the JAVA_HOME environment variable for the WebLogic Server domain:

1. On the machine on which Oracle WebLogic Server is installed, go to the *WebLogic_home***/user_projects/domains/***Domain_Name***/bin** directory.

where:

- *WebLogic_home* is the directory in which you installed the WebLogic Server.
- *Domain_Name* is the domain directory in which PCC will be deployed.

2. Open the **setDomain.sh** file in a text editor.
3. Add or modify the following entries before the **EXTRA_JAVA_PROPERTIES** entry:

```
JAVA_HOME="${JAVA_HOME}"
export JAVA_HOME
```

4. Save and close the file.
5. Restart Oracle WebLogic Server. See the Oracle WebLogic Server documentation for more information.
6. Verify the Java version used by Oracle WebLogic Server. The Java version is displayed when the WebLogic server is started.

Downloading the JDBC Driver for PCC

Download the JDBC driver file and save it in a directory on the machine on which you want to install PCC. Note the path to this directory; you are required to specify this path in the Specify Prerequisite Libraries Location window during the PCC installation. The correct file to download is:

- If you are using Oracle Database version 19c, download the latest version of **ojdbc8.jar** for the 19c database.
- If you are using Oracle Database version 23ai, download the latest version of **ojdbc11.jar** for the 26ai database.

You can download **ojdbc8.jar** or **ojdbc11.jar** from the following location:

https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html

Installing and Configuring BRM

To install and configure BRM:

1. Install BRM. See "Installing BRM" for more information.
2. Set the **PriceDesignCenterInst** business parameter. See "Setting Up the Business Parameter" for more information.

Setting Up the Business Parameter

To use PCC with PDC, you must set up the **PriceDesignCenterInst** business parameter.

To set up **PriceDesignCenterInst**:

1. Go to *BRM_home***/sys/data/config** and run the following command, which creates an editable XML file from the pricing instance of the **/config/business_params** object.

   ```
   pin_bus_params -r BusParamsPricing bus_params_pricing.xml
   ```

   This directory includes the support files used by the **pin_bus_params** utility.

   This command creates the XML file named **bus_params_pricing.xml.out** in your working directory. If you do not want this file in your working directory, specify the path as part of the file name.

2. Search for the following line in the XML file:

   ```
   <PriceDesignCenterInst>disabled</PriceDesignCenterInst>
   ```

   By default, **PriceDesignCenterInst** is set to **disabled**.

3. Change **disabled** to **enabled**.
4. Save the file as **bus_params_pricing.xml**.
5. Use the following command to load this updated file into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_pricing.xml
   ```

6. Read the object with the **testnap** utility or Object Browser to verify that all fields are correct.

Downloading the PCC Software

You can download and install full versions of the PCC software from one of the following locations:

- For BRM 15.*x*.*y* maintenance release: From the Oracle software delivery website (https://edelivery.oracle.com)
- For any 15.*x*.*y*.0.0 patch set release: From the Oracle support website (https://support.oracle.com)

Search for and download the **Oracle Communications Billing and Revenue Management 15.*x*.*y*.0.0** software, where *x* refers to the maintenance release and *y* refers to the patch set release of BRM that you are installing.

The Zip archive includes the PCC installer: **PipelineConfigurationCenter_15.*x*.*y*.0.0_generic.jar**

Installing PCC

Before installing PCC, read the following:

- PCC Installation Overview
- PCC System Requirements
- PCC Preinstallation Tasks

You can install PCC in the following modes:

- **GUI mode:** Use GUI mode when you want to interact with the PCC installer GUI during installation. See "Installing PCC in GUI Mode".
- **Silent mode:** Use silent mode when you install PCC using the same configuration repeatedly. Silent install mode does not use the GUI, and it runs in the background. See "Installing PCC in Silent Mode".

The PCC installer must be launched from the computer hosting the Oracle WebLogic Server domain. PCC installation must be performed by a user who has permissions to write to the Oracle Inventory (**oraInventory**) directory and the *WebLogic_home***/user_projects/domains** directory, where *WebLogic_home* is the directory in which you installed the WebLogic Server.

For more information about the **oraInventory** directory, see the Oracle documentation:

`http://docs.oracle.com`

Installing PCC in GUI Mode

To install PCC in GUI mode:

1. Ensure that the Pipeline Manager database server is running.
2. Start the Oracle WebLogic Server domain administration server and managed server on which you want to deploy PCC.
3. Obtain the PCC software. See "Downloading the PCC Software".
4. To start the PCC GUI installer, go to the directory where you downloaded the PCC software and run the following command:

   ```
   Java_home/bin/java -jar jarFile[ -invPtrLoc invPath/oraInst.loc][ -record -
   destinationFile respPath]
   ```

   where:

   - *Java_home* is the directory in which you installed the latest compatible Java version.
   - *jarFile* is the BRM installer file. For example:

     ```
     PipelineConfigurationCenter_15.x.y.0.0_generic.jar
     ```

   - **-invPtrLoc** starts an optional clause to use if you want to use an **oraInventory** directory that already exists in a different location from the one specified in the **/etc/oraInst.loc** file.
   - *invPath* is the path to the directory in which the **oraInst.loc** file is located.
   - **-record** starts an optional clause to use if you want to create a silent installer response file during the installation.
   - *respPath* is the absolute path to the response file.

   For example:

   ```
   /tools/Linux/jdk/bin/java -jar PipelineConfigurationCenter_15.x.y.0.0_generic.jar
   ```

   or

   ```
   /tools/Linux/jdk/bin/java -jar PipelineConfigurationCenter_15.x.y.0.0_generic.jar -
   invPtrLoc /support/oraInventory/oraInst.loc
   ```

   or

   ```
   /tools/Linux/jdk/bin/java -jar PipelineConfigurationCenter_15.x.y.0.0_generic.jar -
   record -destinationFile /brm15/silent/InstComp.txt
   ```

   or

   ```
   /tools/Linux/jdk/bin/java -jar PipelineConfigurationCenter_15.x.y.0.0_generic.jar -
   invPtrLoc /support/oraInventory/oraInst.loc -record -destinationFile /brm15/silent/
   InstComp.txt
   ```

5. In the Welcome window, click **Next**.
6. If the **oraInst.loc** file is corrupt or not present, the Specify Inventory Directory and Credentials window appears next. Otherwise, go to step 7. In the Specify Inventory Directory and Credentials window, enter the details listed in Table 10-2 and then click **Next**.

Table 10-2 Specify Inventory Directory and Credentials

Field	Description
Inventory Directory	The full path to the inventory directory. This should be an absolute path, not containing environment variables. Do not include the name of the **oraInst.loc** file. The directory and oraInst.loc file will be created if they do not already exist. The default location of the **oraInventory** directory is recorded in the **/etc/oraInst.loc** file.
Operating System Group Name	The name of the operating system group that has write permission to the inventory directory.

7. In the Installation Location window, enter the full path or browse to the directory in which to install PCC and click **Next**.

8. In the Feature Sets Selection window, select **Pipeline Configuration Center** and click **Next**.

9. In the Prerequisite Libraries Location window, enter the full path or browse to the directory in which the JDBC driver is located and lick **Next**. See "Downloading the JDBC Driver for PCC" for more information.

10. In the WebLogic Server Details window, enter the details listed in Table 10-3 for the domain on which you want to deploy PCC. Click **Next**.

Table 10-3 WebLogic Server Details window

Field Name	Description
Host Name	Enter the IP address or the host name of the computer on which the domain is configured.
Port Number	Enter the port number assigned to the domain administration server.
Username	Enter the domain administrator user name.
Password	Enter the password for the domain administrator user.
WebLogic Home	Enter the path of the directory where the Oracle WebLogic Server software is installed. **Note**: The PCC installer does not proceed until it verifies with the running instance of Oracle WebLogic Server that the domain administration server information you entered is valid.
Use SSL?	Do one of the following: • If the WebLogic server does not support SSL, deselect the **Use SSL?** check box. • If the server supports SSL, leave the **Use SSL?** check box selected. When the check box is selected, the **Keystore Type**, **Keystore Location**, and **Keystore Password** fields are mandatory. See "Configuring SSL for the Oracle WebLogic Server Domain" for more information.
Keystore Type	Select the KeyStore certificate type: **PKCS12** or **JKS**.
Keystore Location	Enter or browse to the directory of your client-side KeyStore certificate file, which was generated from the exported public certificate using the **keytool** utility.
Keystore Password	Enter the password for the KeyStore file.

11. In the Target Server window, select the domain administration server or managed server on which you want to deploy PCC. Click **Next**.

> **Note**
>
> Oracle recommends deploying PCC on a managed server. If you select a managed server, ensure that the managed server and the node manager are running.

12. In the Pipeline Manager Database Connection window, enter the details listed in Table 10-4 for connecting to the Pipeline Manager database. Click **Next**.

Table 10-4 Pipeline Manager Database Connection window

Field Name	Description
Host Name	Enter the IP address or the host name of the computer on which the Pipeline Manager database is configured.
Port Number	Enter the port number assigned to the Pipeline Manager database service.
User Name	Enter the user name of the Pipeline Manager database user.
Password	Enter the password of the Pipeline Manager database user.
Service Name	Enter the name of the Pipeline Manager database service. **Note**: The PCC installer will not proceed until it verifies with the running instance of the Pipeline Manager database server that the database information you entered is valid.
SSL Enabled	Select the type of connection required to connect to the database: • **Yes-One Way**: One-way SSL authentication is required. • **Yes-Two Way**: Two-way SSL authentication is required. • **No**: SSL authentication is not required, and the remaining fields are not needed.
Store Type	Select the type of TrustStore and KeyStore file that is used for the SSL connection: **SSO** or **PKCS12**.
Truststore Location	Enter or browse to the directory in which your TrustStore file is located. This file is used for authentication through SSL.
Truststore Password	Enter the password required to access the certificates from the TrustStore.
Keystore Location	Enter or browse to the directory in which your KeyStore file is located. This file is used for authentication through SSL.
Keystore Password	Enter the password required to access the certificates from the KeyStore.

13. In the BRM Connection Details window, enter the details listed in Table 10-5 for connecting to the BRM server. Click **Next**.

Table 10-5 BRM Connection Details window

Field Name	Description
User Name	Enter the user name for connecting to BRM.
Password	Enter the BRM user's password.

Table 10-5 (Cont.) BRM Connection Details window

Field Name	Description
Host Name	Enter the IP address or the host name of the machine on which the primary BRM Connection Manager (CM) or CM Master Process (CMMP) is running.
Port Number	Enter the TCP port number of the CM or CMMP on the host computer. The default value is **11960**.
Service Type	Enter the BRM service type. The default value is **/service/admin_client**.
Service POID ID	Enter the POID of the BRM service. The default value is **1**.
Use SSL?	Do one of the following: • If you have not enabled SSL for BRM, deselect the **Use SSL?** check box. • If you have enabled SSL for BRM, leave the **Use SSL?** check box selected.
Wallet Password	Enter the password for the PCC wallet.
Confirm Wallet Password	Enter the password for the PCC wallet again.

14. In the Pipeline Configuration Center User window, enter the details listed in Table 10-6 for creating a PCC user. Click **Next**.

> **Note**
>
> Entering information in this window is optional. You do not have to create a PCC user now.

Table 10-6 Pipeline Configuration Center User window

Field Name	Description
Username	Enter the name for the PCC user.
Password	Enter a password for the PCC user. **Note:** The PCC user password can contain up to 12 characters and should contain at least one character that is not a letter. In addition, the user name must not be a part of the password.
Confirm Password	Enter the password again.

15. In the Installation Summary window, review the selections you made in the preceding windows, and then click **Install**.

16. The Installation Progress window appears. When installation completes, click **Next.**

> **Note**
>
> After the installation begins, if you click **Stop installation**, the installation process stops, but the files that are already copied are not removed.

17. The Installation Complete window appears. Record PCC URL. You use this URL to access PCC.

> **Note**
>
> If the WebLogic server supports SSL, the secure connection is established for PCC even if you deselect the **Use SSL?** check box in the WebLogic Administration Server Connection window. In this case, two connection URLs are displayed:
>
> - A secured connection URL with SSL listen port and the HTTPS schema
> - An unsecured connection URL with the HTTP schema

18. Click **Finish**.

    The PCC installer exits.

If you did not create a PCC user during installation, you must create one after the installation is complete. See "PCC Postinstallation Tasks" for more information.

See "Verifying the PCC Installation" for information about verifying the successful installation of PCC.

See "About Installation Logs" for information about PCC installer logs.

Installing PCC in Silent Mode

Silent-mode installation enables you to set installation configurations only once and then use those configurations to duplicate the installation on many machines. In this mode, you use a response file template that contains a predefined set of values to install PCC.

> **Note**
>
> The silent installation is not meant for production environments, and it should be used only in development, demonstration, and test environments for setting up quickly or backing up the properties for later use in another non-production environment.

About the Response File Template

The PCC installer includes a response file template, *temp_dir***/pcc/Disk1/stage/Response/oracle.communications.pcc.Complete.rsp**, where *temp_dir* is the directory in which you unzipped the PCC software.

This response file template contains all the parameters that the PCC installer requires during the silent, unattended installation, but it does not contain the parameter values. You must update the response file template per your installation requirements before starting the silent installation.

Creating a Response File

To create a response file:

1. Create the response file by doing one of the following:
   - Create a copy of the response file that was generated during the GUI installation. See "Installing PCC" for more information.

> **Note**
>
> The GUI Installer does not store passwords provided during installation in the response file. You must manually add the passwords after creating a copy of the response file.

- Create a response file using the template by running the following command:

  ```
  Java_home/bin/java -jar jarFile -getResponseFileTemplates
  ```

  where:

  - *Java_home* is the directory in which you installed the latest supported Java version.
  - *jarFile* is the BRM installer file. For example:

    ```
    PipelineConfigurationCenter_15.2.0.0.0_generic.jar
    ```

  A response file is created with the default values.

  You can create as many response files as needed.

2. Open the response file in a text editor.
3. Modify the response file you copied by specifying the key-value information for the parameters you want in your installation.

   > **Note**
   >
   > - The response file template contains guidelines and examples on how to enter the values in the parameters.
   > - The PCC installer treats incorrect context, format, and type values in a response file as if no value were specified.

4. Save and close the response file.

Installing PCC in Silent Mode

To install PCC in silent mode:

1. Create a response file. See "Creating a Response File".
2. Copy the response file you created to the machine on which you are running the silent installation.
3. On the machine on which you are running the silent installation, go to the directory where you downloaded the PCC software, and run the following command:

   ```
   Java_home/bin/java -jar jarFile  -debug -invPtrLoc Inventory_home/oraInventory/
   oraInst.loc [parameter=value] -responseFile path -silent
   ```

   where:

   - *Java_home* is the location of the compatible version of Java.
   - *jarFile* is the BRM installer file. For example:

     ```
     PipelineConfigurationCenter_15.2.0.0.0_generic.jar
     ```

- *path* is the absolute path to the response file.
- *parameter* is the name of an installation parameter.
- *value* is the value of the installation parameter.

For example:

```
Java_home/bin/java -jar PipelineConfigurationCenter_15.2.0.0.0_generic.jar -debug -
invPtrLoc Inventory_home/oraInventory/oraInst.loc INSTALL_TYPE=Complete -
responseFile /response_files/myresponsefile.rsp  -silent
```

The installation runs silently in the background.

4. Open the *PCC_Home***/install/readme.txt** file, where *PCC_Home* is the directory in which you installed PCC.
5. Copy the URL and paste it in your browser's address field.

   You can now access the PCC application.

See "Verifying the PCC Installation" for information on how to verify that the PCC installation was successful.

See "About Installation Logs" for information on the PCC installer logs.

About Installation Logs

The PCC Installer logs can be found in the **oraInventory/logs** directory.

Use the following log files to monitor installations and postinstallation tasks:

- **installAction***TimeStamp***.log**
- **oraInstall***TimeStamp***.err**
- **oraInstall***TimeStamp***.out**
- **silentInstall***TimeStamp***.log** (for silent mode installation)

where *TimeStamp* is the date and time the log file was created.

PCC Postinstallation Tasks

After you install PCC, perform the following postinstallation tasks:

1. Creating a PCC User after Installing PCC
2. Enabling Authentication Cookies
3. Configuring the Session Timeout
4. Configuring SAML 2.0 for SSO Using a Service Provider (Optional)

> **Note**
>
> During installation, the PCC installer copies the PCC **Infranet.properties** file to the WebLogic Server *domain_home* directory. You can update the **Infranet.properties** file in that directory.

Creating a PCC User after Installing PCC

During the PCC installation, the installer creates the **Config Admin** Oracle WebLogic Server group. Users belonging to this group have read and write access to perform all the tasks in PCC.

You can create a PCC user during or after the installation. When you create a user during the installation, the installer automatically adds it to the **Config Admin** parent group.

To create a PCC user after installing PCC, use WebLogic Server Remote Console to create a user and to add the user to the **Config Admin** parent group.

> **Note**
>
> Do not use your browser's remember password feature for the WebLogic Server Remote Console URL. As a precaution, always enter the WebLogic server user name and password manually.

To create a PCC user after installing PCC:

1. Log in to WebLogic Server Remote Console.
2. Click **Security Data Tree** and then **Realms**.

   The Summary of Security Realms page appears.
3. Click the **myrealm** link.

   The myrealm configuration page appears.
4. Click **Authentication Providers** in the tree in the left pane.

   A page with an Authentication Providers table appears.
5. In the Authentication Providers table, select **DefaultAuthenticator**, and then **Users**.
6. Click **New**.
7. Enter a **Name**, **Description**, and **Password** for the user.

   User names must be unique in the security realm and passwords must be eight characters or longer.
8. Click **Create**.
9. In **Membership**, select **Config Admin** parent group from the **Available** section and move it to the **Chosen** section.

   User groups provide an efficient way to manage multiple users with the same responsibilities.

Enabling Authentication Cookies

Oracle recommends deploying PCC only on SSL, which encrypts sensitive data, thus eliminating problems like session stealing.

WebLogic Server enables a user to securely access HTTPS resources in a session that was initiated using HTTP, without loss of session data.

To use secure cookies:

1. Open the *WebLogic_home***/user_projects/domains/***Domain_name***/config/config.xml** file.

   where:

   - *WebLogic_home* is the directory in which you installed WebLogic Server.
   - *Domain_name* is the name of the domain you are configuring.
2. Add **AuthCookieEnabled="true"** to the **<WebServer>** element:

   ```
   <WebServer Name="myserver" AuthCookieEnabled="true"/>.
   ```

> **Note**
>
> By default, **AuthCookieEnabled** is set to **true**. The secured cookies are enabled by default even if the **config.xml** file does not contain this entry. You can set **AuthCookieEnabled** to **false** to disable secured cookies. However, Oracle recommends not disabling it.

You can also set this entry by using WebLogic Remote Console:

1. Log in to WebLogic Remote Console.
2. Click **Edit Tree**, then **Environment**, and then **Domain**.
3. Click the **Web Application** tab.
4. Verify that **Auth Cookie Enabled** is turned on.
5. Click **Save**.

By default, **Auth Cookie Enabled** is turned on, but it is not present in the **config.xml** file. If you turn it off, the **<AuthCookieEnabled>** element is added to the **config.xml** file.

Setting **AuthCookieEnabled** to **true**, the default setting, causes the WebLogic Server instance to send a new secure cookie, _WL_AUTHCOOKIE_JSESSIONID, to the browser when authenticating through an HTTPS connection. After the secure cookie is set, the session is allowed to access other security-constrained HTTPS resources only if the cookie is sent from the browser.

For more information, see the discussion about using secure cookies to prevent session stealing in *Oracle Fusion Middleware Developing Applications with the WebLogic Security Service*.

Oracle recommends keeping cookie settings enabled in the browser. Disabling cookies in the browser disables several features, such as Help.

Configuring the Session Timeout

The default session timeout in PCC is 600 seconds. The WebLogic Server administrator can change this value as follows:

1. Log in to WebLogic Remote Console.
2. Click **Monitoring Tree**, then **Deployments**, and then **Application Management**.

   A page with a list of installed Java EE applications and standalone application modules appears.
3. In the table, click **PipelineConfigurationCenter**.

The information about PipelineConfigurationCenter page appears.

4. Click **Configuration** in the tree in the left pane.
5. Click **Session Descriptor** in the tree in the left pane.
6. In the **Session Timeout (in seconds)** field, enter a new timeout value in seconds.
7. Click **Save**.
8. Return to **Application Management**.
9. In the table, click **PCC-PCCViewController-context-root**.
10. Click the **Configuration** tab.
11. Click **Session Descriptor** in the tree in the left pane.
12. In the **Session Timeout (in seconds)** field, enter the same timeout value, in seconds, that you entered in step 6 .
13. Click **Save**.
14. If you do not already have a deployment plan, or if the deployment plan is unavailable, WebLogic Server creates a deployment plan with the changes above and prompts you to save it. Provide the name and path for the new deployment plan and click **OK**.
15. Click **Application Management** in the tree in the left pane.
16. In the table, select the **PipelineConfigurationCenter** checkbox.
17. Click **Update/Redeploy**.
18. Select **Redeploy - Deployment Source and Plan on Server** and set the **Plan Path** field to the deployment plan.
19. Click **Done**.
20. Restart WebLogic Server.
21. Verify your changes by doing the following:
    a. Log in to WebLogic Remote Console.
    b. Click **Monitoring Tree**, then **Deployments**, and then **Application Management**.

       A page with a list of installed Java EE applications and standalone application modules appears.
    c. In the table, select **PipelineConfigurationCenter**.

       The information about PipelineConfigurationCenter page appears.
    d. Click **Configuration** in the tree in the left pane.
    e. Click **Session Descriptor** in the tree in the left pane.
    f. Verify that **Session Timeout (in seconds)** is set to the value you specified.

For more information, see the discussion about configuring applications for production deployment in *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server*.

Configuring SAML 2.0 for SSO Using a Service Provider

You can use SAML 2.0 for enabling SSO in Pipeline Configuration Center (PCC). SSO allows you to log in to applications using a single user name and password combination.

You can configure SAML authentication in a PCC domain using an Oracle Access Management service provider or an Oracle Identity Cloud Service (IDCS) service provider.

To configure SAML for SSO:

1. Create a SAML2 assertion provider. See "Creating a SAML2 Assertion Provider".
2. Create a SAML2 web single sign-on identity provider partner. See "Creating a SAML2 Web Single Sign-On Identity Provider Partner"
3. Create a SAML2 authenticator. See "Creating a SAML2 Authenticator".
4. Configure SAML2 on the administration server. See "Configuring SAML2 on the Administration Server".
5. Configure a SP **metadata.xml** file for the identity provider. See "Configuring the SP metadata.xml file for the Identity Provider".
6. Create a SAML2 application in IDCS or Oracle Access Management.
7. Update your deployment plan to define the cookie name and path. See "Updating the Deployment Plan of PCC".
8. In your PCC **plan.xml** file, set the SSO_SIGNOUT_URL parameter:

   ```
   SSO_SIGNOUT_URL=IDP's Single Logout URL
   ```

Creating a SAML2 Assertion Provider

To create a SAML2 assertion provider for PCC:

1. Log in to WebLogic Server Remote Console.
2. Click **Edit Tree**, then **Security**, and then **Realms**.

   The Summary of Security Realms page appears.
3. Click the **myrealm** link.

   The myrealm configuration page appears.
4. Click **Authentication Providers** in the tree on the left side.

   A page with an Authentication Providers table appears.
5. Click **New**.

   The Create a New Authentication Provider page appears.
6. In the **Name** field, enter **samlPCCAsserter**.
7. From the **Type** list, select **SAML2IdentityAsserter**.
8. Click **Create**.
9. Restart WebLogic Server.

Creating a SAML2 Web Single Sign-On Identity Provider Partner

To create a SAML2 web single sign-on identity provider partner

1. Log in to WebLogic Server Remote Console.
2. Click **Security Data Tree** and then **Realms**.

   The Summary of Security Realms page appears.
3. Click the **myrealm** link.

   The myrealm configuration page appears.
4. Click **Authentication Providers** in the tree on the left side.

A page with an Authentication Providers table appears.

5. In the table, select **samlPCCAsserter**.

   The configuration page for samlPCCAsserter appears.

6. Click **Partners** in the tree on the left side.
7. Click **New**.

   The Create a new Identity Provider Partner page appears.

8. In the **Name** field, enter **WebSSO-IdP-Partner**.
9. From the **Type** list, select **Web Single Sign-On Identity Partner**.
10. In the **Meta Data File Name** field, enter the name and the path to the XML file that contains the identity provider's metadata.
11. Click **Create**.
12. Click **WebSSO-IdP-Partner** at the tree on the left side.
13. Click the **General** tab.
14. In the **General** tab, turn on **Enabled**, **Virtual User**, and **Process Attributes**.
15. In the **Redirect URIs** field, enter **/pcc/***.
16. Click **Save**.

Creating a SAML2 Authenticator

To create a SAML2 authenticator for PCC:

1. Log in to WebLogic Server Remote Console.
2. Click **Edit Tree**, then **Security**, and then **Realms**.

   The Summary of Security Realms page appears.

3. Click the **myrealm** link.

   The myrealm configuration page appears.

4. Click **Authentication Providers** in the tree on the left side.

   A page with an Authentication Providers table appears.

5. Click **New**.
6. In the **Name** field, enter **samlPCCAuthenticator**.
7. From the **Type** list, select **SAMLAuthenticator**.
8. From the **Control Flag** list, select **SUFFICIENT.**
9. Click **Create**.
10. In the Authentication Provider table, arrange the providers in the following order using the **Move Down** and **Move Up** buttons.
    - **Trust Service Identity Asserter**
    - **samlPCCAuthenticator**
    - **samlPCCAsserter**
    - **DefaultAuthenticator**
    - **DefaultIdentityAsserter**

11. Restart the Weblogic Server.

Configuring SAML2 on the Administration Server

To configure SAML 2.0 on the administration server:

1. Log in to WebLogic Server Remote Console.
2. Click **Edit Tree**, then **Environment**, and then **Servers**.

   The Summary of Servers page appears.
3. In the Servers table, click the administration server.

   A page containing settings for the administration server appears.
4. Click the **Security** subtab, and then the **SAML 2.0 General** subtab.
5. In the **Published Site URL** field, enter **http://***PCC_hostname:port***/saml2**.

   where:

   - *PCC_hostname* is either the PCC application host name or the load balancer host name.

     port is the port on which PCC is listening on.
6. In the **Entity ID** field, enter **samlPCCAsserter**.
7. Click **Save**.
8. Click the **SAML 2.0 Service Provider** subtab.
9. Turn on **Enabled**.
10. Turn on **POST Binding Enabled**.
11. From the **Preferred Binding** list, select **HTTP/POST**.
12. In the **Default URL** field, enter **http://***PCC_hostname:port***/pcc/login.html**.

    where:

    - *PCC_hostname* is the PCC application host name or load balancer name.
    - *port* is the port on which PCC is listening on.
13. Click **Save**.
14. Restart WebLogic Server.

Configuring the SP metadata.xml file for the Identity Provider

To configure SP **metadata.xml** file for identity provider:

1. Log in to WebLogic Server Remote Console.
2. Click **Monitoring Tree**, then **Environment**, and then **Servers**.

   The Summary of Servers page appears.
3. In the Servers table, click the administration server.

   A page containing settings for the administration server appears.
4. Click **SAML 2.0** subtab.
5. Click **Publish metadata**.
6. In the **File Name** field, enter the full path and the name of the file.

7. Click **Done**.

Updating the Deployment Plan of PCC

To update the deployment plan of PCC:

1. Merge the following contents with your existing PCC deployment plan:

```
<?xml version='1.0'encoding='UTF-8'?>
<deployment-plan
xmlns="http://xmlns.oracle.com/weblogic/deployment-plan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"xsi:schemaLocation="http://
xmlns.oracle.com/weblogic/deployment-plan http://xmlns.oracle.com/weblogic/
deployment-plan/1.0/deployment-plan.xsd"global-variables="false">
  <application-name>PCC</application-name>
  <variable-definition>
   <variable>
    <name>cookie-name</name>
    <value>JSESSIONID</value>
   </variable>
   <variable>
    <name>cookie-path</name>
    <value>/pcc</value>
   </variable>
  </variable-definition>
  <module-override>
   <module-name>PCCView.war</module-name>
   <module-type>war</module-type>
   <module-descriptor external="true">
    <root-element>weblogic-web-app</root-element>
    <uri>WEB-INF/weblogic.xml</uri>
    <variable-assignment>
     <name>cookie-name</name>
     <xpath>/weblogic-web-app/session-descriptor/cookie-name</xpath>
     <operation>replace</operation>
    </variable-assignment>
    <variable-assignment>
     <name>cookie-path</name>
     <xpath>/weblogic-web-app/session-descriptor/cookie-path</xpath>
     <operation>remove</operation>
    </variable-assignment>
   </module-descriptor>
  </module-override>
</deployment-plan>
```

2. Log in to WebLogic Server Remote Console.
3. Click **Monitoring Tree**, then **Deployments**, and then **Application Management**.

   A page with a list of installed Java EE applications and standalone application modules appears.
4. In the table, select **pcc#1.0**.
5. Click **Update/Redeploy** and do one of the following:
   - If your updated deployment plan is located on the server containing WebLogic Server, select **Update Deployment Plan on Server** and enter the location of the file in the **Plan Path** field.
   - If your updated deployment plan is located on your local machine, select **Update Deployment Plan on Local Machine** and click Choose File ⬆ to browse to the file or enter the file name in the **Plan** field.

6. Click **Done**.

Verifying the PCC Installation

Verify whether PCC was installed successfully by doing the following:

- Checking the State of the Installed Components
- Logging In to PCC
- About File Permissions

Checking the State of the Installed Components

You can verify that PCC is installed correctly by checking the state of the installed components.

To check the state of the installed components:

1. Log in to the PCC domain in WebLogic Remote Console.
2. Click **Monitoring Tree**, then **Deployments**, and then **Application Management**.

   A page with a list of installed Java EE applications and standalone application modules appears.
3. If PCC is installed correctly, the deployed **PipelineConfigurationCenter** application appears in the **Active** state.

Logging In to PCC

You can log in to PCC to verify that PCC is installed.

To log in to PCC:

1. In a supported browser, navigate to the URL provided by the PCC installer at the end of the installation.

   The PCC login page appears.

   > **Note**
   >
   > The URL to access PCC is also available in the *PCC_Home***/install/readme.txt** file, where *PCC_Home* is the directory in which you installed PCC.

2. Enter the user name and password in the login window for PCC.
3. Click **Login**.

   The PCC home page appears, verifying that PCC is successfully installed.

About File Permissions

- The following default permissions are set for the installed files:
  - `rw-------` 600 (for all non-executable files)
  - `rwx------` 700 (for all executable files)

Permissions are set to the most restrictive level, and the WebLogic Server administrator can add or revoke permissions. Oracle recommends keeping the permissions as restrictive as possible.

- Protect the WebLogic configuration (JMS, JDBC, and so on) file, **config.xml**, in the domain's **config** directory with proper permissions.

Verifying Configuration Entries

The installer stores the configuration entries in the PCC wallet. These entries are used to control the connections from Java applications to BRM.

After PCC has been installed on each machine, check the configuration entries stored in the PCC wallet. You can verify the entries by doing the following:

> **Note**
>
> If PCC and the BRM server are installed on separate machines, copy the PCC wallet to the machine in which the BRM server is installed and perform the following procedure on the machine in which the BRM server is installed.

1. Go to the directory in which you installed PCC, and source the **source.me** file:

   Bash shell:

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

   C shell:

   ```
   source source.me.csh
   ```

2. Run the following command:

   ```
   orapki wallet display -wallet PCC_Wallet_Path
   ```

   where *PCC_Wallet_Path* is the path to the PCC wallet.

   The entries stored in the client wallet appear.

   If the entries do not appear, you must store the entries manually in the PCC wallet by using the **pin_crypt_app** utility.

To secure this connection information, encrypt the password by using the **pin_crypt_app** utility. See "About Encrypting Passwords" in *BRM Developer's Guide*.

Uninstalling PCC

> **Note**
>
> After uninstalling PCC, you must manually undeploy the PCC application deployed on the WebLogic server domain. See the discussion about undeploying applications in the Oracle WebLogic Server documentation.

To uninstall PCC:

1. Go to the *PCC_home***/oui/bin** directory.
2. Run the following command from a GUI (for example, VNC server):

   ```
   ./deinstall.sh
   ```

   The Distribution to Uninstall window appears.
3. Select the components you want to uninstall.
4. Click **Uninstall**.

   The Welcome window appears.
5. Click **Next**.

   The Uninstallation Summary window appears.
6. Click **Uninstall**.

   The Uninstallation Progress window appears.
7. Click **Next**.

   The Uninstallation Complete window appears.
8. Click **Finish**.

The PCC installer removes all files except log files. You must delete the log files manually. The log files are in the **oraInventory/logs** directory.

Uninstalling PCC in Silent Mode

To uninstall PCC in silent mode:

1. Go to the *PCC_home***/oui/bin** directory.
2. Run the following command:

   ```
   ./deinstall.sh -responseFile path -silent
   ```

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

   A message indicating that PCC has been uninstalled successfully appears.

Deleting Files after Uninstalling PCC

The following files remain in the system after PCC is uninstalled:

- Install logs

  The log files are in the **oraInventory/logs** directory.
- *PCC_home***/oui/data.properties**

  This file is used to auto-populate the data during reinstalls.

Delete these files manually if you do not need them, or protect them appropriately if they are required for further installations.

By default, these files are created with the file-level permission 640 (owner can read/write; group members can read; others cannot do anything).

Administering PCC

A PCC administrator is responsible for the day-to-day tasks of maintaining and managing PCC and its users. You perform the following tasks as a PCC administrator:

- Managing PCC Security
- Monitoring PCC
- Managing PCC

Managing PCC Security

Oracle WebLogic Server includes a security architecture that provides a secure foundation for applications. PCC depends on the WebLogic Server security framework to secure its resources and servers. Managing PCC security involves the following:

- Managing security realms: Configuring new security realms, changing the default security realm, and deleting security realms
- Managing users and groups: Defining users and assigning them to a group that can be authenticated in a security realm
- Managing security providers: Managing security providers that provide security services to applications to protect WebLogic Server resources

See Oracle WebLogic Remote Console Online Help for more information on managing security realms, managing users and groups, and managing security providers.

Monitoring PCC

Regularly monitoring your system ensures fast recognition and resolution of problems or issues. You can use WebLogic Remote Console to monitor the following:

- PCC domain
- PCC administration server and managed servers

See Oracle WebLogic Remote Console Online Help for more information.

Managing PCC

You manage PCC by managing the WebLogic server on which PCC is installed. This includes starting and stopping the PCC administration server and managed servers.

See Oracle WebLogic Remote Console Online Help for more information.

Troubleshooting PCC

Although PCC installation is designed to be trouble free, you might encounter a problem. You can troubleshoot PCC installation problems using information in the following sections:

- Troubleshooting Checklist
- Using Error Logs to Troubleshoot PCC
- Diagnosing PCC Problems
- Getting Help for PCC Problems

Troubleshooting Checklist

When problems occur, it is best to do some troubleshooting before you contact Oracle support:

- You know your installation better than Oracle support does. You know whether anything in the system has been changed, so you are more likely to know where to look first.
- Troubleshooting skills are important. Relying on Oracle support to research and solve all your problems prevents you from being in full control of your system.

If you have a problem with your PCC system, ask yourself the following questions because Oracle support will ask them:

- What exactly is the problem? Can you isolate it?

  Oracle support needs a clear and concise description of the problem, including when it began to occur.
- What do the log files say?

  This is the first thing that Oracle support asks for. Check the error log for the PCC component you are having problems with.
- Has anything changed in the system? Did you install any new hardware or new software? Did the network change in any way? Does the problem resemble another one you had previously? Has your system usage recently jumped significantly?
- Is the system otherwise operating normally? Has response time or the level of system resources changed? Are users complaining about additional or different problems?

Using Error Logs to Troubleshoot PCC

PCC error log files provide detailed information about system problems. If you are having a problem with PCC, look in the log files.

PCC records information about actions performed in the PCC GUI in Oracle WebLogic Server log files. See Oracle WebLogic Remote Console Online Help for more information.

Diagnosing PCC Problems

PCC problems can be diagnosed using the Oracle WebLogic Server Diagnostic Framework, which enables you to collect, archive, and access diagnostic information about applications hosted on the WebLogic server. See Oracle WebLogic Remote Console Online Help for more information.

This section lists some common PCC problems and describes how to diagnose the error messages and resolve the problems.

Refreshing the Browser

Do not refresh your browser.

Refreshing the browser causes unpredictable problems in PCC.

Unable to Perform Any Task after Logging In

After you log in to PCC, the links in the navigation pane are not displayed, and you cannot perform any task.

This means that you are not in the **Config Admin** WebLogic Server group. Contact your system administrator, and request to be added to that group.

Getting Help for PCC Problems

If you cannot resolve a PCC problem, contact Oracle support.

Before you contact Oracle support, try to resolve the problem with the information in the log files. See "Using Error Logs to Troubleshoot PCC" for more information. If this does not help resolve the problem, gather the following information:

- A clear and concise description of the problem, including when it began to occur.
- Relevant portions of the relevant log files.
- Relevant configuration files.
- Recent changes in your system after which the problem occurred, even if you do not think they are relevant.
- A list of all PCC components and patches installed on your system.

When you are ready, report the problem to Oracle support.

Secure Deployment Checklist

To deploy PCC securely, follow this checklist:

1. Preinstallation steps:
   a. Enable SSL for the target Oracle WebLogic Server domain.
   b. Configure the server KeyStore certificate, and get the client KeyStore trusted certificate.
   c. Configure Oracle Database advanced security encryption and integrity algorithms for a secure connection from the installer.
   d. Verify that you have the correct version of the JDK with the latest security update installed and configured with your PCC/WebLogic Server installation.
2. Installation steps:
   a. Select the SSL mode, select the KeyStore type, and provide the client KeyStore certificate (**.p12** [recommended] or **.jks** file) for connecting to a WebLogic server over SSL.
3. Postinstallation steps:
   a. If you do not need the installation log files, delete them.
   b. The WebLogic Server administrator must create PCC users based on the roles and privileges.
   c. Do not use your browser's remember password feature for the WebLogic Remote Console URL.
   d. Enable secure cookies.
   e. Verify that file permissions for the installed files are 600 for all files that are not executable and 700 for all executable files.
4. Uninstallation steps:

a. Delete the log files in the **oraInventory/logs** directory manually if you do not need them, or protect them appropriately if they are required for further installations.

11 Installing BRM REST Services Manager

Learn how to use the Oracle Communications Billing and Revenue Manager (BRM) REST Services Manager package to install both the BRM REST Services Manager API and the BRM REST Services Manager SDK on your system.

Topics in this document:

- Downloading the BRM REST Services Manager Package
- Installing BRM REST Services Manager
- BRM REST Services Manager Postinstallation Tasks
- Troubleshooting SSL Connection Errors

For more information about the BRM REST Services Manager API and SDK, see *REST Services Manager API for Billing and Revenue Management*.

Downloading the BRM REST Services Manager Package

You can download and install the BRM REST Services Manager software from one of the following locations:

- For BRM 15.2.0 or any 15.2.*x* maintenance release: From the Oracle software delivery website (https://edelivery.oracle.com)
- For any 15.2.*x*.*y* patch set release: From the Oracle support website (https://support.oracle.com)

Search for and download the **Oracle Communications Billing and Revenue Management 15.2.*x*.*y*.0** software, where *x* refers to the maintenance release and *y* refers to the patch set release of BRM that you are installing.

The Zip archive includes the BRM REST Services Manager installer: **BRM_REST_Services_Manager-15.2.*x*.*y*.0.jar**

Installing BRM REST Services Manager

The BRM REST Services Manager package installs both the BRM REST Services Manager API and the BRM REST Services Manager SDK on your system.

You can install BRM REST Services Manager in the following modes:

- **GUI mode**: Use GUI mode when you want to interact with the BRM REST Services Manager installer GUI during installation. See "Installing BRM REST Services Manager in GUI Mode".
- **Silent mode**: Use silent mode when you install BRM REST Services Manager using the same configuration repeatedly. Silent install mode does not use the GUI, and it runs in the background. See "Installing BRM REST Services Manager in Silent Mode".

Installing BRM REST Services Manager in GUI Mode

To install BRM REST Services Manager in GUI mode:

1. Go to the *temp_dir* directory and run one of these commands:
   - To start the GUI installer:

     ```
     Java_home/bin/java -jar jarFile
     ```

     where:
     - *Java_home* is the directory in which you installed the latest compatible Java version.
     - *jarFile* is the BRM installer file. For example:

       ```
       BRM_REST_Services_Manager-15.2.0.0.0.jar
       ```

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

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

     where *FilePath* is the path to the directory in which the **oraInst.loc** file is located.
   - To start the GUI installer and create a silent installer response file during the installation:

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

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

   > **Tip**
   >
   > You can run the following command to get more details about the options.
   >
   > ```
   > Java_home/bin/java -jar jarFile -help
   > ```

2. In the Welcome window, click **Next**.

   > **Note**
   >
   > If the **oraInst.loc** file is corrupt or not present, the Installation Inventory window appears next. Otherwise, the Installation Location window appears.

3. (Optional) In the Installation Inventory window, enter the details listed in Table 11-1 and then click **Next**.

   Table 11-1   Installation Inventory

Field	Description
Inventory Directory	The full path of the inventory directory. The default **oraInventory** directory is located in the **/etc/oraInst.loc** (Linux) file.

Table 11-1 (Cont.) Installation Inventory

Field	Description
Operating System Group	The name of the operating system group that has write permission to the inventory directory.

4. In the Installation Location window, enter the full path or browse to the directory in which you want to install BRM REST Services Manager and then click **Next**.
5. In the Feature Sets Selection window, select the components to install, deselect any components that you do not want to install, and then click **Next**.

> **Note**
>
> You cannot deselect a component if it is required to install any of the selected components.

If you are installing an optional component for the first time after an upgrade, and you see an error similar to the following, see "Problem: An Error Occurred When Selecting an Optional Component to Install".

```
The distribution Billing Revenue Management 15.2.0.0.0 contains incompatible
features with the following:
Billing Revenue Management 15.0.1.0.0 [CORE 15.2.0.0.0->CORE 15.0.1.0.0 (CORE
15.2.0.0.0->[CORE 15.0.1.0.0])]
```

6. In the Server Details window, enter the details listed in Table 11-2 for the server on which you want to deploy BRM REST Services Manager and then click **Next**.

Table 11-2 Server Details

Field	Description
Host Name	The host name of the computer on which the server is configured.
Port Number	The port number available on the host.
SSL Port Number	The SSL port number on which you want to install BRM REST Services Manager.
KeyStore Location	The path of the client-side KeyStore file you generated.
KeyStore Password	The password used for accessing the KeyStore file.

7. In the Base URL Details window, enter the base URL to include in the response of BRM REST Services Manager requests. Click **Next**.
8. In the BRM Connection Details window, enter the details listed in Table 11-3 for connecting to the BRM server and then click **Next**.

Table 11-3 BRM Connection Details

Field	Description
User Name	The user name for connecting to BRM.
Password	The password of the BRM user.

Table 11-3 (Cont.) BRM Connection Details

Field	Description
Host Name	The host name of the computer on which the primary BRM Connection Manager (CM) or CM Master Process (CMMP) is running.
Port Number	The TCP port number of the CM or CMMP on the host computer. The default value is **11960**.
Service Type	The BRM service type. The default value is **/service/pcm_client**.
Service POID Id	The POID of the BRM service. The default value is **1**.
Use SSL?	Do one of the following: • If you have not enabled SSL for BRM, deselect this check box. • If you have enabled SSL for BRM, select this check box.
Wallet Password	The password for the BRM wallet.
Confirm Wallet Password	Enter the password for the BRM wallet again.

9. In the Security Details window, do one of the following and then click **Next**.
   - If you want to configure BRM REST Services Manager securely, select **Yes**.
   - If you want to configure BRM REST Services Manager in a test installation mode, select **No** and go to step 11.
10. In the Identity Provider Details window, enter the details listed in Table 11-4 and then click **Next**.

Table 11-4 Identity Provider Details

Field	Description
Identity URL	The base URL of your Identity Provider (IdP) server.
Scope Audience	The primary audience registered for the application which is appended for scopes.
Audience	The name of the Oracle Access Manager OAuth server.
Client Id	The client ID of the application.
Client Secret	The client secret of the application.

11. In the Installation Summary window, review the selections you made in the preceding windows and then click **Install**.
12. The Installation Progress window appears. When the installation completes, click **Next**.

> **Note**
>
> After the installation begins, you cannot stop or cancel the installation.

13. The Installation Complete window appears. Make note of the BRM REST Services Manager base URL. You use this URL to access BRM REST Services Manager.

> **Note**
>
> You can find the BRM REST Services Manager logs in the *REST_home*/**logs** directory, where *REST_home* is the BRM REST Services Manager installation directory.

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

    The BRM REST Services Manager installer exits.

15. (For Security Enabled with Identity Provider) If your IdP requires an introspection endpoint to validate the JWT, uncomment the **introspect-endpoint-uri** entry in your **application.yaml** file and set it to the introspection endpoint.

    > **Note**
    >
    > For example, if you are using Oracle Access Management as your IdP, you need to uncomment and set the **introspect-endpoint-uri** entry.

16. (For Security Enabled with Oracle Access Management Only) If you have not yet migrated to the latest versions of Oracle Access Management, where roles and groups are included in the JSON Web Tokens (JWTs) token, do the following:

    > **Note**
    >
    > This step is necessary when your JWTs do not adhere to the MicroProfile JWT RBAC v2.1 specification and lack a "groups" claim because it enables fetching user or client groups and roles. If your JWTs conform to the JWT RBAC v2.1 specification, skip this step.

    a. In your **application.yaml** file, uncomment the following **oam-role-mapper** entries.

    ```
    - oam-role-mapper:
         oud:
              host-name:
              admin-user-name:
              http-port:
              https-port:
              users-base-dn:
              groups-base-dn:
              msgType:
    urn:ietf:params:rest:schemas:oracle:oud:1.0:SearchRequest
              filter: (&(objectclass=*)
    (uniqueMember=cn=__USER_NAME__,__USER_BASE_DN__))
    ```

    b. Update the entries for your environment:

    - **host-name**: The Oracle Unified Directory host name.
    - **admin-user-name**: The administration user name for the Oracle Unified Directory.
    - **http-port**: The Oracle Unified Directory HTTP port.
    - **https-port**: The Oracle Unified Directory HTTPS port.
    - **users-base-dn**: The Oracle Unified Directory user domain name.
    - **groups-base-dn**: The Oracle Unified Directory group domain name.

- **msgType**: The message type based on the schema used to search roles in the Oracle Unified Directory.
- **filter**: The filter based on the user attribute.

> **Note**
>
> The **admin-password** key is the administration password for the Oracle Unified Directory. Do not set this entry directly. Instead, write the password to the wallet as OAM_OUD_ROOTUSERPASS. To store the client secret password in the wallet, enter the following command:
>
> ```
> java -cp
> ".;oraclepkijarLocation;cetjarLocation"com.portal.cet.ConfigEditor -
> setconf -wallet clientWalletLocation –parameter configEntry -value value
> ```

17. (For Security Enabled with Oracle Identity Cloud Service Only) If IDCS is not configured to include groups or roles in JSON Web Tokens (JWTs), do the following:

> **Note**
>
> This step is necessary when your JWTs do not adhere to the MicroProfile JWT RBAC v2.1 specification and lack a "groups" claim because it enables fetching user or client groups and roles. If your JWTs conform to the JWT RBAC v2.1 specification, skip this step.

a. In your **application.yaml** file, uncomment the following **idcs-role-mapper** entries:

```
- idcs-role-mapper:
     multitenant: false
     oidc-config:
        client-id:
     identity-uri:
        cache-config:
        cache-timeout-millis: 10000
```

b. Update the entries for your environment:

- **client-id**: The client ID generated by the Identity Server, used to validate the token.
- **identity-uri**: The URI of the Identity Server, used as the base URL to retrieve metadata from the Identity Server.

> **Note**
>
> The **client-secret** key is for the client secret generated by the Identity Server, used to authenticate the application when requesting a JWT based on a code. Do not set this entry directly. Instead, write the client secret password to the wallet as CLIENT_SECRET. To store the client secret password in the wallet, enter the following command:
>
> ```
> java -cp
> ".;oraclepkijarLocation;cetjarLocation"com.portal.cet.ConfigEditor -
> setconf -wallet clientWalletLocation -parameter configEntry -value value
> ```

18. After installation completes, run the *REST_home***/scripts/start-brm-rsm.sh** script to bring up BRM REST Services Manager.

Installing BRM REST Services Manager in Silent Mode

The silent installation uses a response file in which you have set installation information. To obtain the response file, you run the GUI installer for the first install. The GUI installer generates a response file that contains the key-value pairs based on the values that you specify during the GUI installation. You can then copy and edit the response file to create additional response files for installing the BRM REST Services Manager on different machines.

Creating a Response File

To create a response file:

1. Create a copy of the response file that was generated during the GUI installation. See "Installing BRM REST Services Manager in GUI Mode" for more information.

   > **Note**
   >
   > A response file was created only if you ran the GUI installer with this command:
   >
   > ```
   > Java_home/bin/java -jar jarFile -record -destinationFile path
   > ```

2. Open the response file in a text editor.
3. Manually add all passwords to the response file. The GUI Installer does not store the passwords that you provided during installation in the response file.
4. Modify the key-value information for the parameters you want in your installation.

   > **Note**
   >
   > The Installer treats incorrect context, format, and type values in a response file as if no value were specified.

5. Save and close the response file.

Installing BRM REST Services Manager in Silent Mode

To install BRM REST Services Manager in silent mode:

1. Create a response file. See "Creating a Response File".
2. Copy the response file you created to the machine on which you run the silent installation.
3. On the machine on which you run the silent installation, go to the *temp_dir* directory and run the following command:

   ```
   Java_home/bin/java -jar jarFile -debug -invPtrLoc Inventory_home/oraInventory/
   oraInst.loc -responseFile path -silent
   ```

   where:

   - *Java_home* is the directory in which you installed the latest supported Java version.
   - *jarFile* is the BRM installer file. For example:

     ```
     BRM_REST_Services_Manager-15.2.0.0.0.jar
     ```

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

   For example:

   ```
   Java_home/bin/java -jar BRM_REST_Services_Manager-15.2.0.0.0.jar -debug -invPtrLoc
   Inventory_home/oraInventory/oraInst.loc -responseFile /tmp/
   BRM_REST_Services_Manager.rsp -silent
   ```

   The installation runs silently in the background.

   > **Tip**
   >
   > You can run the following command to get more details about the options.
   >
   > ```
   > Java_home/bin/java -jar jarFile -help
   > ```

4. After installation completes, run the *REST_home***/scripts/start-brm-rsm.sh** script to start BRM REST Services Manager.
5. Open the BRM REST Services Manager URL (**https://***hostname***:***port***/**) in a browser.

   > **Note**
   >
   > You can find the port number in the *REST_home***/scripts/application.yaml** file. The HTTPS port is in the **server.port** entry, and the HTTP port is in the **server.sockets.plain.port** entry.

BRM REST Services Manager Postinstallation Tasks

After you install BRM REST Services Manager, perform these postinstallation tasks:

1. Setting Up a Custom TrustStore
2. Configuring Policies for API Authorization
3. Excluding the BRM REST Services Manager SDK
4. Updating BRM Connection Details
5. Updating the Base URL
6. Modifying BRM REST Services Manager Properties

7. Setting Up Zipkin Tracing in BRM REST Services Manager
8. Connecting BRM REST Services Manager to Oracle Analytics Publisher
9. Configuring BRM REST Services Manager for High Availability
10. Allowing the Use of Deprecated Cyphers
11. Configuring and Adding Custom Mapper Files
12. Customizing REST Services with the Mapper Framework
13. Configuring the REST Services Manager Notification Service

Setting Up a Custom TrustStore

If you want to use a custom TrustStore instead of the Java TrustStore, do the following:

1. Stop BRM REST Services Manager by running the *REST_home***/scripts/stop-brm-rsm.sh** script.
2. In the *REST_home***/scripts/brm_rsm.properties** file, configure the path to the TrustStore file in the TRUST_STORE_FILE_PATH key.
3. Store the TrustStore passphrase in the TRUST_STORE_PASSPHRASE configuration entry in the client wallet by running the following command:

   ```
   java -cp ".;oraclepkijarLocation;cetjarLocation"com.portal.cet.ConfigEditor -setconf
   -wallet clientWalletLocation –parameter configEntry -value value
   ```

   where:

   - *clientWalletLocation* is the full path to the client wallet.
   - *value* is the TrustStore passphrase in Base64-encoded format.
4. Start BRM REST Services Manager by running the *REST_home***/scripts/start-brm-rsm.sh** script.

Configuring Policies for API Authorization

To configure the policies for API authorization:

1. Stop BRM REST Services Manager by running the *REST_home***/scripts/stop-brm-rsm.sh** script.
2. Define the API authorization rules in a policy file.

   You can use the sample authorization policy file (*REST_home***/scripts/authorization-policy.yaml**) as a template for defining API authorization rules.
3. For any new BRM REST Services Manager API endpoints, ensure that appropriate policy statements are added to the file. This is essential for enforcing proper authorization and access restrictions for each new API.
4. Start BRM REST Services Manager by running the *REST_home***/scripts/start-brm-rsm.sh** script.

Excluding the BRM REST Services Manager SDK

The BRM REST Services Manager package installs both the BRM REST Services Manager API and the BRM REST Services Manager SDK on your system.

If you do not want to include the BRM REST Services Manager SDK in your deployment, remove this entry from the *REST_home***/scripts/brm_rsm.properties** file: **EXTENSION_LIB_PATH**.

Updating BRM Connection Details

By default, the BRM REST Services Manager installer stores sensitive information such as passwords in the Oracle wallet, and BRM REST Services Manager retrieves the passwords from the Oracle wallet. However, you can update this sensitive information after the installation.

To update the BRM connection details:

1. Stop BRM REST Services Manager using this command:

   ```
   REST_home/scripts/stop_brm_rsm.sh
   ```

2. Navigate to the *REST_home***/wallet** directory, where *REST_home* is the BRM REST Services Manager installation directory.
3. Run the following command to update the BRM connection details:

   ```
   java -cp ".;oraclepkijarLocation;cetjarLocation"com.portal.cet.ConfigEditor -setconf
   -wallet clientWalletLocation –parameter configEntry -value value
   ```

   where:

   - *clientWalletLocation* is the path to the client wallet.
   - *configEntry* is the configuration entry in the client wallet.
   - *value* is the appropriate value for the respective entry in the client wallet.

4. Start BRM REST Services Manager using this command:

   ```
   REST_home/scripts/start_brm_rsm.sh
   ```

Updating the Base URL

After installation, you can update the base URL with the resource details to return in the response of BRM REST Services Manager requests. To do so, edit the *REST_home***/scripts/application.yaml** file.

Modifying BRM REST Services Manager Properties

To modify the BRM REST Services Manager properties:

1. Stop BRM REST Services Manager using this command:

   ```
   REST_home/scripts/stop_brm_rsm.sh
   ```

2. Open the *REST_home***/scripts/brm_rsm.properties** file in a text editor.
3. Modify the properties based on your requirements. For example, changing the port number and log level.
4. Save the **brm_rsm.properties** file.
5. Start BRM REST Services Manager using this command:

   ```
   REST_home/scripts/start_brm_rsm.sh
   ```

Setting Up Zipkin Tracing in BRM REST Services Manager

You can trace the flow of API calls made to BRM REST Services Manager by using Zipkin, which is an open-source tracing system. For more information, see the Zipkin website: https://zipkin.io/.

To set up Zipkin tracing for BRM REST Services Manager:

1. Install Zipkin. See the Zipkin Quickstart documentation: https://zipkin.io/pages/quickstart.html.
2. Stop BRM REST Services Manager using this command:

   ```
   REST_home/scripts/stop_brm_rsm.sh
   ```

3. In your **application.yaml** file, update the following entries, as necessary:

   ```
   otel:
      traces:
         exporter: zipkin
      sdk:
         disabled: true
      service:
         name: brm-rest-service-manager
      exporter:
         zipkin:
            endpoint: ${traces.protocol}://${traces.host}:${traces.port}/api/v2/spans
   ```

4. Start BRM REST Services Manager using this command:

   ```
   REST_home/scripts/start_brm_rsm.sh
   ```

Afterward, you can start tracing the flow of API calls to BRM REST Services Manager by using the Zipkin UI or Zipkin API.

Connecting BRM REST Services Manager to Oracle Analytics Publisher

To enable BRM REST Services Manager to retrieve PDF invoices generated by Oracle Analytics Publisher:

1. Update the following Oracle Analytics Publisher connection details in the BRM REST Services Manager wallet:
   - **BIP_USERID**: Set this to the Oracle Analytics Publisher user that has web access.
   - **BIP_PASSWORD**: Set this to the password for the Oracle Analytics Publisher user password.
   - **BIP_URL**: Set this to the URL for accessing the Oracle Analytics Publisher instance. For example: **http://***hostname***:***port***/xmlpserver/services/PublicReportService_v11**.
2. To store the connection details in the wallet, enter the following command:

   ```
   java -cp ".;oraclepkijarLocation;cetjarLocation"com.portal.cet.ConfigEditor -setconf
   -wallet clientWalletLocation –parameter configEntry -value value
   ```

Configuring BRM REST Services Manager for High Availability

You can optionally configure BRM REST Services Manager for high availability.

Because BRM REST Services Manager is a relatively simple API service, you can support high availability by using the open-source load balancer of your choice, such as Apache HTTP

Server or HAProxy. The load balancer automatically routes requests to whichever server is available, which enables greater scalability by sharing the traffic among servers, and also lets processing continue if an instance is down.

The basic steps for configuring BRM REST Services Manager for high availability are:

1. Install and configure BRM REST Services Manager. You can install multiple instances on the same server by using a different port or install different instances on different servers.
2. Install a load balancer. You can install the load balancer on the same server as one of your BRM REST Services Manager instances, or a different server with access to the BRM REST Services Manager servers.
3. Add the details of the BRM REST Services Manager servers to the load balancer, using a proxy to pass traffic to the group of servers.

Allowing the Use of Deprecated Cyphers

According to security best practices, by default BRM Rest Services Manager restricts less-secure cyphers from being used. These restricted cyphers are listed in *REST_home*/**scripts/security.properties**. You can change this configuration if you are willing to take the risk of allowing less-secure cyphers.

To allow individual restricted cyphers:

1. Stop BRM REST Services Manager using this command:

   ```
   REST_home/scripts/stop_brm_rsm.sh
   ```
2. Back up the *REST_home*/**scripts/security.properties** file.
3. Open the *REST_home*/**scripts/security.properties** file in a text editor.
4. Remove the names of any cyphers you would like to allow.
5. Save the **security.properties** file.
6. Start BRM REST Services Manager using this command:

   ```
   REST_home/scripts/start_brm_rsm.sh
   ```

To remove all cypher restrictions:

1. Stop BRM REST Services Manager using this command:

   ```
   REST_home/scripts/stop_brm_rsm.sh
   ```
2. Open the *REST_home*/**scripts/brm_rsm.properties** file in a text editor.
3. Set the value of the **OVERRIDE_SECURITY_PROPERTIES** property to **false**.
4. Save the **brm_rsm.properties** file.
5. Start BRM REST Services Manager using this command:

   ```
   REST_home/scripts/start_brm_rsm.sh
   ```

Customizing REST Services with the Mapper Framework

Learn how to customize REST service payload mapping for Oracle Communications REST Services Manager using the Mapper Framework in a cloud native environment. You can use the mapper file to control how REST API payloads are transformed to and from internal BRM data structures (FLIST). By customizing the mapper file, you can extend or override out-of-the-box mappings, adapt to business requirements, and enable advanced field transformations.

Topics in this document:

- Introduction to the Mapper File
- Understanding the Mapper File Structure
- Examples and Supported Field Attributes
- Mapper Features
- Best Practices for Customizing Mapper Files

Introduction to the Mapper File

The mapper file defines the transformation logic for each REST API endpoint and the event framework. It maps fields in incoming JSON requests to BRM FLIST fields and maps FLIST response fields back to outgoing JSON. Customizing the mapper file provides flexibility without requiring changes to underlying service code.

You can use the mapper file to:

- Implement custom REST API payload logic.
- Extend or override default field mappings.
- Support multiple API versions and endpoints.
- Add custom business rules, data validation, and type conversions.

Sample Events Configuration:

```
config:
        paths:
          event:
            v5:
              ProductCreateEvent:
                mapper-specification-keys:
                  default: Product
                  Event:
                    execute:
                    - "mapper.brm.v5.tmf637.productCreateEvent.execute"
                    request:
                    - "mapper.brm.v5.tmf637.productCreateEvent.request"
                    response:
                    - "mapper.brm.v5.tmf637.productCreateEvent.response"
                  Product:
                    request:
                    - "mapper.brm.v5.tmf637.post.request.product"
                    - "mapper.brm.v5.tmf637.post.request.extendedProduct"
                    response:
                    - "mapper.brm.v5.tmf637.post.response.product"
                    - "mapper.brm.v5.tmf637.post.response.extendedProduct"
```

where,

- **ProductCreateEvent** defines the type of event.
- **mapper-specification-keys** specifies logical profiles or mapping variants.
- **default** is used when no explicit profile is provided.
- **execute**, **request**, and **response** designate the mapping configuration files to be applied for execution opcode, request transformation, and response transformation respectively.

Understanding the Mapper File Structure

The mapper file defines the transformation logic for each REST API endpoint. It maps fields in incoming JSON requests to BRM FLIST fields and maps FLIST response fields back to outgoing JSON. Customizing the mapper file provides flexibility without requiring changes to underlying service code.

You can use the mapper file to:

- Implement custom REST API payload logic.
- Extend or override default field mappings.
- Support multiple API versions and endpoints.
- Add custom business rules, data validation, and type conversions.

Mapper files are written in YAML format and contain these key sections:

- **request**: Defines how incoming JSON fields are mapped to BRM FLIST fields.
- **response**: Defines how FLIST response fields are mapped to outgoing JSON.
- **execute**: Contains BRM opcode and opcode flag information.

> **Note**
>
> Mapper files do **not** contain a top-level **validation** section. Use **optional** on a per-field basis as needed.

Examples and Supported Field Attributes

The order in which mapper keys are defined determines the order of overrides: if a key is defined last, it will override any common keys present earlier. You must ensure that your custom mapper keys are referenced correctly in your configuration (such as in **application.yaml** or through deployment values).

Example configuration snippet:

```
config:
  paths:
    apiSpecifications:
      brm/productInventory/v5/product:
        post:
          mapper-specification-keys:
            default: Product
            Product:
              execute:
                - "mapper.brm.v5.tmf637.post.execute.product"
              request:
                - "mapper.brm.v5.tmf637.post.request.product"
                - "mapper.brm.custom.v5.tmf637.post.request.product"
              response:
                - "mapper.brm.v5.tmf637.post.response.product"
                - "mapper.brm.custom.v5.tmf637.post.response.product"
```

where,

- **config** is the top-level configuration object containing all Mapper Framework settings.

- **paths** is the section that defines REST API endpoint path specifications.
- **apiSpecifications** is the mapping of API specifications to their endpoints.
- **brm/productInventory/v5/product** is the specific API resource path being mapped.
- **post** is the HTTP POST method being specified for mapping.
- **mapper-specification-keys** is the list specifying which mappers to use for the endpoint and method.
- **default** is the fallback mapping key used if no specific match for schema type is found from the request payload.
- **Product** is the schema type for the request and response mapping obtained from request payload.
- **request** is the listing of request-mapping keys or files to handle incoming API payload mapping.
- **response** is the listing of response-mapping keys or files to handle returned API payload mapping.

Example 1: Simple Mapping

```
request:
  product:
    Product:
      field: PIN_FLD_PRODUCT
      optional: false
      dataFields:
        id:
          field: PIN_FLD_POID
          fieldValue: ""
          optional: false
response:
  product:
    PIN_FLD_PRODUCT:
      field: product
      type: array
      dataFields:
        PIN_FLD_POID:
          field: id
          type: string
```

where,

- **request** is the top-level section defining how incoming JSON is mapped to BRM fields.
- **Product** is the logical identifier/object being mapped in the request or response.
- **field** is the target field within the payload or mapping output/input location.
- **dataFields** is the grouping of child fields requiring mapping under the main object.
- **id** is the attribute or subfield (such as a product identifier) being mapped.
- **PIN_FLD_POID** is the BRM internal field for Portal Object Identifier (the mapped field in BRM).

Example 2: Array and Substructure Mapping

```
request:
  product:
    orderItems:
      field: PIN_FLD_ORDER_ITEMS
```

```
          optional: false
          arrayIndex: 0
          dataFields:
            price:
              field: PIN_FLD_PRICE
              fieldValue: ""
              optional: truee
```

where,

- **orderItems** is an array field within the product object, representing line items in the order.
- **PIN_FLD_ORDER_ITEMS** is the BRM array field for order items; square brackets indicate an array.
- **arrayIndex** specifies the array position used for mapping (for example, zero is the first element).
- **price** is the field representing the price for each order item.
- **PIN_FLD_PRICE** is the BRM field for price, the mapping destination for the price data.

Example 3: Dynamic FieldValue Calculation

```
fieldValue: "${(getVal(context[3], 'PIN_FLD_BASE_AMOUNT') + getVal(context[3],
'PIN_FLD_FEE')) * getVal(context[2], 'PIN_FLD_MULTIPLIER') - getVal(context[5],
'PIN_FLD_DEDUCTION')}"
```

where,

- **PIN_FLD_PRICE** is the BRM field for price, where the calculated value is stored.
- **fieldValue** is the value assigned to this field; in this case, it uses Jakarta Expression Language (EL).
- **getVal** is a Jakarta EL function that retrieves the value of a given field from the specified context/hierarchy level.
- **context** is an array representing the hierarchical path through the payload or FLIST object at different levels; context[n] refers to a specific depth in the hierarchy.
- **base**, **supportFee**, **multiplier**, and **deduction** are field names within the corresponding context objects used in the calculation formula.

Mapper Features

The Mapper Framework allows you to map and transform REST API payloads to and from BRM data structures. You can use these features to handle complex mapping scenarios.

API-Based Mapping

You can define mapping rules for each REST API endpoint and HTTP method, tailoring the mapping logic to each business operation. This enables you to ensure each API's request and response format properly aligns with the required BRM data structure. For example, you can configure separate mappings for **/productInventory/v5/product** for POST and GET methods, ensuring the request and response structures match each API's requirements.

Example:

```
mapper:
    brm:
        v5:
            tmf637:
                post:
                    execute:
                        product:
                            brmOpcode: PRODUCT_INVENTORY_POST
                            opcodeFlag: 0
                    request:
                    response:
                get:
                    execute:
                        product:
                            brmOpcode: PRODUCT_INVENTORY_GET
                            opcodeFlag: 0
                    request:
                    response:
```

Versioning Support

You can create and define separate mappings for different versions of APIs. You can use the corresponding mappings based on the version of the API being invoked.

Example:

```
mapper:
    brm:
        v5:
            tmf637:
                productCreateEvent:
                productDeleteEvent:
                productAttributeValueChangeEvent:
                productStateChangeEvent:
                delete:
                post:
                patch:
                get:
        v6:
            tmf637:
                productCreateEvent:
                productDeleteEvent:
                productAttributeValueChangeEvent:
                productStateChangeEvent:
                delete:
                post:
                patch:
                get:
```

Opcode, Schema, and Flags

Each mapping can specify a BRM opcode. The opcode determines which internal BRM operation runs for a request. You can also attach schema information to control FLIST structure validation, and apply flags to influence how the opcode is executed.

> **Note**
>
> The BRM opcode supports Jakarta expressions. You can use Jakarta expression for opcode evaluation aswell

Example:

```
{
    "operation": "edit",
    "@type": "Product",
    "suspendedBatchObjs": [
        {
            "suspendedBatchObj": "0.0.0.1+-product+181321"
        },
        {
            "suspendedBatchObj": "0.0.0.1+-product+129321"
        },
        {
            "suspendedBatchObj": "0.0.0.1+-product+101321"
        }
    ]
}
```

Sample Expression:

```
brmOpcode: "${getVal(context[0], 'operation') == 'edit' ? 'SUSPENSE_SEARCH_EDIT' :
(getVal(context[0], 'operation') == 'recycle' ? 'SUSPENSE_SEARCH_RECYCLE' :
(getVal(context[0], 'operation') == 'writeoff' ? 'SUSPENSE_SEARCH_WRITE_OFF' :
'SUSPENSE_SEARCH_DELETE'))}"
```

In this case, the **brmOpcode** is determined depending on the value of the **operation** field in the request payload.

Request and Response Mapping

Easily map REST API JSON fields to corresponding BRM FList fields for both requests and responses. You can handle simple or complex (nested, array) structures. Each mapping file contains a **request** block and a **response** block.

- In the **request** block, you define how fields in the incoming JSON map to BRM FLIST fields.
- In the **response** block, you map BRM FLIST fields back to JSON for the API output.

Mappings can be simple (one field to one field), or complex, including arrays and nested structures.

Example:

```
request:
  productName:
    field: PIN_FLD_NAME
    optional: false
response:
  PIN_FLD_NAME:
    field: productName
    type: string
```

Field Enumeration

Field enumeration is a way to restrict a field’s value to a specific set of allowed options. When you define an enumeration for a field, only the listed values are considered valid, and any other input will be rejected during validation. This ensures data consistency between API payloads and BRM, and helps prevent errors from invalid or unexpected values. In the following example, you can use different string values from request and to convert it to corresponding values accepted by BRM. For example, **Credit Card** changes to **10001** for FList population. This is applicable for both request and response.

Example (Enumeration):

```
payType:
  field: PIN_FLD_PAY_TYPE
  fieldValue: "${{'Credit Card': 10001, 'Cash': 10011, 'Check': 10012}
[getMapKey(context[0], 'payType')]}"
```

Validation Rules for Map

For the **validation** key in a request, you can use the following rules:

- To specify the type of a field, use one of the supported types: string, int, double, boolean, object, array, long, date.

  ```
  validation: "type:[string]"
  ```

- To specify a date format (the format must be readable by Java):

  ```
  validation: "format:[yyyy-MM-dd]"
  ```

- For enumerated values, define a list of permissible values:

  ```
  validation: "enum:[value1,value2,value3]"
  ```

- To restrict the range of any numerical field:

  ```
  validation: "range:[min=1,max=1000]"
  ```

- To use regex pattern matching on a field:

  ```
  validation: "pattern:[^[a-zA-Z0-9]+$]"
  ```

- To use multiple validation rules at once, provide them as a comma-separated list:

  ```
  validation: "range:[min=1000,max=5000],pattern:[^\\d+$],enum:[3000,2000,2048]"
  ```

> **Note**
>
> Enclose all rule values within square brackets **[ ]**, as shown in the examples above.

Mapper Expressions (Jakarta EL)

You can use Jakarta Expression Language (EL) within **field** and **fieldValue** to evaluate values at runtime in your mapping files. These expressions use context objects to refer to fields at different levels of the JSON or FLIST structure.

You can use these built-in expressions inside the **${...}** blocks in your YAML mapping files to reference fields, perform lookups, and create URLs during mapping.

Mapper Expression Functions
The Mapper Framework supports three primary mapper expression functions by default:

- **getVal**: Use **getVal** to extract the value of a specific key from a JSON object or FList. In request mappings, **getVal(context[index], 'key')** retrieves a field from the JSON payload. In response mappings, it fetches a value from the FList object.

  ```
  fieldValue: "${getVal(context[0], 'accountId')}"
  ```

- **getMapKey**: Use **getMapKey** for field enumeration and value mapping. It looks up the appropriate BRM code or value based on a key from the context and an inline map definition.

  ```
  fieldValue: "${{'Credit Card': 10001, 'Cash': 10011, 'Check':10012}
  [getMapKey(context[0], 'payType')]}"
  ```

- **generateResourceUrl**: Use **generateResourceUrl** to generate an href or URL reference for a BRM resource. This is commonly used to set the href field in API responses.

```
    href:
      fieldValue: "${generateResourceUrl('paymentMethods/v4/paymentMethod/')}"
```

Object Support (Request and Response)

The Object parameter supports two types of objects for both request and response processing:

- **context** is a list that stores data based on hierarchy. You must specify the index to indicate which level of the object you are referring to. If you do not reference an index, **context** at the same level where the expression exists will be used.
  - On the request side, **context** contains **JsonNode** objects representing the request payload.
  - On the response side, **context** contains **FList** objects representing the response payload.
- **rootContext** stores the complete payload during processing and you can use it directly without specifying an index.
  - On the request side, **rootContext** is a **JsonNode** object that holds the entire request payload.
  - On the response side, **rootContext** is an **FList** object that holds the entire response payload.

When you process a bundled product, the system creates a new hierarchy. **rootContext** allows you to access data outside the current context. Once processing is complete, it restores the original hierarchy. You may use **context[index]**, **context**, or **rootContext** as the Object.

Understanding the Hierarchy
Consider this request mapping:

```
productPrice:
  field: PIN_FLD_PRODUCT_PRICE
  validation: ""
  optional: true
  dataFields:
    description:
      field: ""
      fieldValue: "ProductPrice"
      validation: ""
      optional: true
    priceAlteration:
      field: PIN_FLD_VALUES
      validation: ""
      optional: true
      dataFields:
        _attype:
          field: ""
          fieldValue: "PriceAlteration"
          validation: ""
          optional: true
        price:
          field: ""
          validation: ""
          optional: true
          dataFields:
            dutyFreeAmount:
              field: PIN_FLD_PRICING_INFO
              validation: ""
              optional: true
              dataFields:
                unit:
                  field: PIN_FLD_UNIT_STR
                  validation: ""
                  optional: true
                value:
                  field: PIN_FLD_AMOUNT
                  validation: ""
                  optional: true
            taxIncludedAmount:
              field: ""
              fieldValue: ""
              validation: ""
              dataFields:
                unit:
                  field: PIN_FLD_UNIT_STR
                  validation: ""
                  optional: true
                value:
                  field: PIN_FLD_AMOUNT
                  validation: ""
                  optional: true
```

In this mapping:

- **context[0]** holds the entire payload.
- At any **dataFields** of **dutyFreeAmount**, the hierarchy objects are:
  - **context[0]** is the entire JSON payload as **JsonNode**
  - **context[1]** is the **productPrice** as **JsonNode**
  - **context[2]** is the **price** as **JsonNode**
  - **context[3]** is the **dutyFreeAmount** as **JsonNode**
- For **dataFields** of **taxIncludedAmount**, the hierarchy objects are:
  - **context[0]** is the entire JSON payload as **JsonNode**
  - **context[1]** is the **productPrice** as **JsonNode**
  - **context[2]** is the **price** as **JsonNode**
  - **context[3]** is the **taxIncludedAmount** as **JsonNode**

The response side maintains a similar hierarchy structure.

Removing Inherited or Obsolete Fields

You can override or delete fields inherited from a base mapping by specifying **action: delete** for that field in your custom mapper file. Example:

```
obsoleteField:
  action: delete
```

This removes the inherited mapping for **obsoleteField** in your endpoint configuration.

Advanced Mapping Scenarios

You can use the features listed above for solutions to common mapping scenarios:

Handling Data Fields
You map deeply nested structures using the **dataFields** attribute. Example:

```
account:
  field: PIN_FLD_ACCOUNT
  dataFields:
    firstName:
      field: PIN_FLD_FIRST_NAME
    lastName:
      field: PIN_FLD_LAST_NAME
```

This maps **account.firstName** and **account.lastName** in the payload to the appropriate BRM fields.

If a base field is nested, you can omit the keyword **dataFields** from a substruct or array field in the base mapper file to use the base schema and that if a nested field is encountered, it is an instance of the same object. For example:

```
product:
    field: PIN_FLD_PRODUCTS
    fieldValue: ""
    optional: false
    validation: ""
```

In this case, the schema applicable for the **product** field is assumed to be the same as that used for the base request, and is a nested field of the same schema. This is applicable for both the request and the response.

Assigning Default Values

You provide a default value with **fieldValue**. If the source data is missing, REST Services Manager uses this value. Example:

```
quantity:
  field: PIN_FLD_QUANTITY
  fieldValue: "1"
  optional: true
```

If the API payload omits **quantity**, the field will default to 1.

> **Note**
>
> If there is Jakarta EL expression is provided in the **fieldValue** attribute, it is evaluated. However, a string value is treated as a default value.

Changing BRM Field Values

You use **fieldValue** or mapper expressions (Jakarta EL) to set or transform BRM field values based on input, context, or calculations. Example:

```
price:
  field: PIN_FLD_PRICE
  fieldValue: "${getVal(context[1], 'base') * 1.07}"
```

This multiplies the base price by 1.07 (such as to apply tax).

Mapping Arrays and Array Indexes

You use **arrayIndex** to place or retrieve data at a specific index in a BRM array field. Example:

```
aliases:
  field: PIN_FLD_ALIAS_LIST
  arrayIndex: 0
  dataFields:
    name:
      field: PIN_FLD_NAME
```

This places the **name** value into the first element of the BRM alias list.

Handling Empty or Optional Fields

You use **optional: true** to indicate a field may be absent in the request or response. If the field exists but is empty, REST Services Manager omits it to the BRM FLIST unless prevented by additional validation. Example:

```
nickname:
  field: PIN_FLD_NICKNAME
  optional: true
```

Handling href Fields

You generate URL links (href) for resources in the response using static values or by running helper functions, allowing you to provide links back to created or referenced resources. Example:

```
href:
  field:
  fieldValue: "${generateResourceUrl('productInventory/v5/product/')}"
  type: string
```

Class Reference for Response Payloads:

To define fields such as **@type** and **@referredType** which are not present anywhere in FList but are required in the json fiel to reference classes:

```
_attype:
    field: ""
    type: string
    fieldValue: "Product"
```

where,

- **type** is a string value.
- **fieldValue** is the class name to populate into response payload.
- **_attype** is the defned key, where **_at** is used instead of @ to prevent YAML syntax issues.

> **Note**
>
> These classes mentioned above are for reference only; the actual data is taken from request payload.

Specifying Data Types in Response

You control data type in the response using the **type** key—for instance, mapping an integer to a string or a date. The supported key types are:

- **date**
- **string**
- **float**
- **double**
- **integer**
- **boolean**

- **json**

Example:

```
PIN_FLD_START_T:
  field: startDate
  type: date
```

Using Dot Notation

You reference nested fields in request or response mappings using dot notation, such as **parentField.childField**. Example:

```
"contact.email":
  field: PIN_FLD_CONTACT_INFO.PIN_FLD_EMAIL
```

Handling Query Parameters, Path Parameters, and Storable Classes

You access HTTP query or path parameters by including their mapped names as top-level fields in your mapping. For advanced scenarios requiring additional object types or extended models (**storable class**), use custom handlers and reference them in your mapping definitions.

Example (Query Parameter):

```
queryParameter:
    field: PIN_FLD_ARGS
    dataFields:
         id:
             field: PIN_FLD_IS_BUNDLE
             fieldValue: ""
             validation: ""
```

Example (Path Parameter):

```
pathParameter:
  field: PIN_FLD_ARGS
  dataFields:
    id:
      field: PIN_FLD_POID
```

Example (StorableClass):

```
paymentMethod:
  storableClass: "/payinfo"
```

> **Note**
>
> Query Parameters, Path Parameters, and Storable Classes are only applicable for GET.

Mapping Child Data from Unmapped Parent Objects

When there's an object/array in the JSON payload but there's no corresponding FList to map it to, yet the data inside the object/array is required for FList creation. You can handle this as follows:

```
price:
  field: PIN_FLD_PRICING_INFO
  validation: ""
  optional: true
  dataFields:
    _attype:
      field: ""
      fieldValue: "Price"
      validation: ""
      optional: true
    dutyFreeAmount:
      field: ""
      validation: ""
      optional: true
      dataFields:
        unit:
          field: PIN_FLD_UNIT_STR
          validation: ""
          optional: true
        value:
          field: PIN_FLD_AMOUNT
          validation: ""
          optional: true
```

Sample JSON:

```
{
  "price": {
    "@type": "Price",
    "dutyFreeAmount": {
      "unit": "USD",
      "value": 29.99
    }
  }
}
```

Sample FList:

```
0 PIN_FLD_PRICING_INFO    SUBSTRUCT [0] allocated 20, used 2
1     PIN_FLD_AMOUNT        DECIMAL [0] 29.99
1     PIN_FLD_UNIT_STR      STR [0] "USD"
```

Splitting and Mapping Composite JSON Fields to Multiple BRM Fields

There can be a case where you need to pass multiple BRM fields and multiple Jakarta EL expressions for a JSON entry. For this case, you can pass a comma-separated list of values as shown in the example.

```
contact:
  field: PIN_FLD_NAMEINFO
  fieldValue: ""
  validation: ""
  dataFields:
    contactName:
      field: PIN_FLD_FIRST_NAME, PIN_FLD_MIDDLE_NAME, PIN_FLD_LAST_NAME
      fieldValue: "${getVal(context, 'contactName').split(' ')[1]!=null ?
getVal(context, 'contactName').split(' ')[0]: null}, ${getVal(context,
'contactName').split(' ')[2]!=null ? getVal(context, 'contactName').split(' ')[1] :
null}, ${getVal(context, 'contactName').split(' ')[2] != null ? getVal(context,
'contactName').split(' ')[2] : (getVal(context, 'contactName').split(' ')[1] != null  ?
getVal(context, 'contactName').split(' ')[1] : getVal(context, 'contactName').split(' ')
[0])}"
      validation: ""
```

JSON PATCH Features

When you prepare a JSON Patch, you form a JSONPath expression and include it in the JSON Patch payload. The payload is an array of operations. Each object in the array contains these attributes:

- **op**: Specifies the operation. Possible values are:
  - **replace**: Replaces the value at the specified path with a new value.
  - **add**: Adds a value at the path.
  - **remove**: Removes the attribute at the path.
- **path**: Contains the JSONPath expression for the operation.
- **value**: Specifies the value for the "replace" and "add" operations.

JSONPath Expressions

lists the supported JSONPath expressions and describes how each one is used to access or modify elements within a JSON payload.

Table 11-5 Rules for Writing JSONPath Expressions

Expression	Description
$.	The expression starts with **$.** to refer to the root of the payload.
$.status	Refers to the **status** key in the root object.
$.productOffering.id	Refers to the **id** property inside the **productOffering** object.
$.productCharacteristic[?(@.name=='ServiceAlias' && @.id=='235' && @.valueType=='array')]	Selects elements from the **productCharacteristic** array that match all specified conditions. Use conditional statements inside square brackets to filter elements.
?(@.name=='ServiceAlias' && @.id=='235' && @.valueType=='array')	This is a conditional subexpression. It starts with a **?** and uses **@.** before JSON keys. Only **==** and **&&** operators are supported in JSON Patch. Enclose the condition in parentheses.
$.productCharacteristic[?(@.name=='ServiceAlias' && @.id=='235' && @.valueType=='array')].value	Accesses the **value** attribute of the selected **productCharacteristic** element.

Best Practices for Customizing Mapper Files

- Validate the YAML syntax before deployment to prevent runtime errors.
- Use ConfigMaps for mapping files with a size limit upto 1MiB in Kubernetes. Use SDK for larger files.
- Always increment **restartVersion** for new mappings. After changing the restartVersion, run the helm upgrade command to force restart.
- Review REST Services Manager pod logs for troubleshooting.
- Use versioning and modular mappings for backward compatibility and gradual migration.
- Ensure key ordering in configuration yields intended override results.

Configuring and Adding Custom Mapper Files

By default, BRM REST Services Manager uses predefined mapper files to translate between JSON payload objects and BRM opcode flists for the latest versions of account and inventory management. However, you can customize this mapping by creating and adding your own mapper files. This allows you to extend or override the default mappings to meet your specific integration needs.

To extend or override the mapping between BRM REST Services Manager JSON objects and BRM opcode flists:

1. Create YAML files containing your custom mapping definitions. You can review sample files in the *REST_home*/**referenceMappers** directory for guidance.
2. Copy your custom YAML files to *REST_home*/**extendedMappers**.
3. Update the BRM REST Services Manager configuration so your custom YAML files are loaded at runtime. Open the *REST_home***/scripts/application.yaml** file. Provide the path from your custom mapper files as requests, responses, or execute respectively. Update the **application.yaml**:

   ```
   config:
     paths:
       event:
         v5:
           ProductCreateEvent:
             mapper-specification-keys:
               default: Product
               Event:
                 execute:
                   - "mapper.brm.v5.tmf637.productCreateEvent.execute"
                 request:
                   - "mapper.brm.v5.tmf637.productCreateEvent.request"
                 response:
                   - "mapper.brm.v5.tmf637.productCreateEvent.response"
                   - "mapper.custom.v5.tmf637.productCreateEvent.response"
               Product:
                 request:
                   - "mapper.brm.v5.tmf637.post.request.product"
                   - "mapper.brm.v5.tmf637.post.request.extendedProduct"
                   - "mapper.custom.v5.tmf637.post.request.product"
                 response:
                   - "mapper.brm.v5.tmf637.post.response.product"
                   - "mapper.brm.v5.tmf637.post.response.extendedProduct"
                   - "mapper.custom.v5.tmf637.post.response.product"
   ```

   where,

   - **ProductCreateEvent** defines the type of event.
   - **mapper-specification-keys** specifies logical profiles or mapping variants.
   - **default** is used when no explicit profile is provided.
   - **execute**, **request**, and **response** designate the mapping configuration files to be applied for execution opcode, request transformation, and response transformation respectively.

   > **ⓘ Note**
   >
   > For more information on custom mapper keys, see "Adding BRM REST Services Manager Keys" in the *BRM Cloud Native Deployment Guide*

4. Restart BRM REST Services Manager to load your changes. Run:

   ```
   REST_home/scripts/stop_brm_rsm.sh
   REST_home/scripts/start_brm_rsm.sh
   ```

Configuring the REST Services Manager Notification Service

Configure the REST Services Manager Notification Service and validate your deployment. See the relevant topics for more information.

- (Optional) Review the overview and prerequisites before proceeding with configuration. See "Overview and Prerequisites for REST Services Manager Notification Service" for more information.
- Follow the REST Services Manager Notification Service configuration process to ensure all components are set up correctly. See "REST Services Manager Notification Service Configuration Process" for more information.
- Set up the required connection profiles for secure integration. See "Setting Up Connection Profiles" for more information.
- Complete any outstanding post-installation tasks as described. See "Completing Post-Installation Tasks" for more information.
- Perform verification and sanity checks to ensure your installation is functional. See "Performing Verification and Sanity Checks" for more information.
- Apply the recommended security best practices before moving to production. See "Security Best Practices for REST Services Manager Notification Service" for more information.

Overview and Prerequisites for REST Services Manager Notification Service

Before you begin configuration, ensure that your environment satisfies all prerequisites for the REST Services Manager Notification Service. This section lists the required Kafka infrastructure, Oracle database wallet setup, and configuration file requirements.

> **Note**
>
> In order to get event notifications, uncomment or add the required configurations as shown in this document.

Kafka Cluster and Topic requirements

- Provision a Kafka cluster with sufficient brokers to meet your throughput and high availability requirements.
- Define all topics required for the REST Services Manager Notification Service, including source, sink, retry, and dead-letter queue (DLQ) topics.
- Ensure topics are created with the desired number of partitions and that appropriate access control lists (ACLs) are applied.
- Record the hostnames and port numbers of all Kafka brokers.
- Monitor Kafka server logs from time to time to validate if the connections are established.

Oracle Database Wallet Setup for Sensitive Values

- Configure an Oracle database wallet to securely store sensitive credentials, such as SSL passphrases, keystore/truststore files, and database schema credentials.
- The population of the Oracle database wallet with security certificates is dependent on the database listener and instance being pre-configured for SSL/TLS. Ensure that the

database encryption layer is active before attempting to map Keystore and Truststore locations within the wallet.

- Store the necessary wallet files and keystore/truststore files for secure connection to the database in a location secured with appropriate file permissions.
- Do not place passwords or secrets in plain configuration files.

> **Tip**
>
> Maintain separate folders for keystores, truststores, and the Oracle database wallet.

> **Note**
>
> Review the default *REST_home***/scripts/application.yaml** supplied with the deployment package and familiarize yourself with the configuration block structure.

REST Services Manager Notification Service Configuration Process

This section outlines the end-to-end workflow for configuring and setting up the Notification Service. Steps include extracting delivery packages, editing configuration files, and securely applying SSL/SASL parameters are needed for Kafka broker connectivity.

To configure and deploy Kafka, Oracle database consumers, and kafka producers:

1. Open the *REST_home***/scripts/application.yaml** file for editing.
2. Un-comment the sections relevant to your deployment:
   - Specify the active connection profiles.
   - Specify the active consumer and producer profiles.
3. Populate connection parameters for Kafka and any required databases.
4. Apply the appropriate SSL/SASL parameters to secure broker access.
5. Store secrets in the wallet using a **key-value** format. Use the corresponding keys within your configuration to retrieve the passwords at runtime.

   ```
   java -cp ".;oraclepkijarLocation;cetjarLocation" com.portal.cet.ConfigEditor -
   setconf -wallet clientWalletLocation –parameter configEntry -value value
   ```
6. Save the configuration and verify all folder paths referenced in *REST_home***/scripts/application.yaml** exist and are accessible by the REST Services Manager process.
7. Start REST Services Manager application to start the notification services.

To apply broker SSL/SASL parameters and store passphrases securely:

- Enter SSL configurations (keystore, truststore locations and types, passwords) in *REST_home***/scripts/application.yaml**, using wallet references.
- Do not write plain credentials in configuration files. Reference them via wallet lookup.

> **Note**
>
> A response file was created only if you ran the GUI installer with this command:
>
> *Java_home*/**bin/java -jar** *jarFile* **-record -destinationFile** *path*

Setting Up Connection Profiles

Configure the Kafka and database connection profiles by mapping related parameters and integrating with your wallet-based secrets. Guidance covers property definitions and secure reference of credential values in the configuration.

Kafka Connection Profile

- Set security protocol such as **SSL** or **SASL_SSL** under the *properties* tag.
- Assign locations and types for keystore and truststore files.
- Specify credentials via wallet or environment variable references.

Example block for Kafka connection profiles in the *REST_home***/scripts/application.yaml**

```
connection-profiles:
     - id: "cp1"
       profile-type: "kafka"
       host: "host"
       port: port
       username: ""
       password-key: "CP1_PASSWORD_WALLET_KEY"
       truststore-type: PKCS12
       truststore-location: "/home/truststore.p12"
       truststore-password-key: "CP1_TRUSTSTORE_WALLET_KEY"
       keystore-type: PKCS12
       keystore-location: "/home/keystore.p12"
       keystore-password-key: "CP1_KEYSTORE_WALLET_KEY"
       properties:
         auto.offset.reset: latest
         enable.auto.commit: false
         session.timeout.ms: 60000
         heartbeat.interval.ms: 10000
         connections.max.idle.ms: 3600000
         max:
           poll:
             records: 10
             interval.ms: 600000
           partition.fetch.bytes: 1048576
         acks: all
         retries: 3
         batch.size: 16384
         linger.ms: 1
         buffer.memory: 33554432
         compression.type: none
```

where,

- **id** is the unique identifier for the connection profile, referenced by consumers or producers to select this profile.
- **profile-type** is the type of connection; set to "kafka" for Kafka connections.
- **host** is the address of the Kafka broker to connect to.

- **port** is the network port used to connect to the Kafka broker.
- **username** is the authentication username, if required for SASL or similar authentication mechanisms (often left empty for SSL-only setups).
- **password-key** is the key referencing the password (e.g., stored in the Wallet) for authenticating this profile.
- **truststore-type** is the type of the truststore file (usually PKCS12 or JKS) used for SSL/TLS connections.
- **truststore-location** is the path to the truststore file containing trusted certificate authorities.
- **truststore-password-key** is the key to retrieve the truststore password (e.g., stored in the Wallet), ensuring secure access.
- **keystore-type** is the type of the keystore file (usually PKCS12 or JKS) holding client SSL certificates.
- **keystore-location** is the path to the keystore file containing the client's SSL certificate and private key.
- **keystore-password-key** is the key to retrieve the keystore password (e.g., stored in the Wallet), ensuring secure access.
- **properties** is an object containing additional Kafka client properties (such as offsets, timeouts, batching, etc.). It follows standard kafka properties so the key and value must be as per standard kafka documentation guidelines.
  - **auto.offset.reset** determines where to start consuming if no offset is committed ("latest" starts from the latest message).
  - **enable.auto.commit** specifies if the consumer should auto-commit offsets; set this to **false** for manual management. (Always set this to false in REST Services Manager.)
  - **session.timeout.ms** sets the timeout in milliseconds to detect consumer failures.
  - **heartbeat.interval.ms** sets how often to send heartbeats to the Kafka broker.
  - **connections.max.idle.ms** sets the maximum idle time for a connection before closing.
  - **max.poll.records** sets the maximum number of records returned in a single poll.
  - **max.poll.interval.ms** sets the maximum interval between poll calls before the broker considers the consumer failed.
  - **max.partition.fetch.bytes** controls the maximum bytes per partition fetched in a request.
  - **acks** defines the number of acknowledgments the producer requires for a request (e.g., "all" for full commit).
  - **retries** is the maximum number of retry attempts for failed produce requests.
  - **batch.size** sets producer batch memory size in bytes.
  - **linger.ms** delays sending records to allow batching (in milliseconds).
  - **buffer.memory** is the total producer buffer memory in bytes.
  - **compression.type** sets the compression algorithm to use ("none", "gzip", etc.).

Example block for database connection profiles in the *REST_home***/scripts/application.yaml**

```
- id: "cp2"
  profile-type: "database"
  host: "host"
  port: port
```

```
       truststore-type: PKCS12
       truststore-location: "/home/truststore.p12"
       truststore-password-key: "CP2_TRUSTSTORE_WALLET_KEY"
       keystore-type: PKCS12
       keystore-location: "/home/keystore.p12"
       keystore-password-key: "CP2_KEYSTORE_WALLET_KEY"
       properties:
         service-name: pindb.example.com
```

where,

- **service-name** is the specific name of the database service (or SID) to which the connection should be established.

Setting up Consumers

Example block for kafka consumers in the *REST_home***/scripts/application.yaml**

```
consumers:
     - connection-profile-id: "cp1"
       properties:
         group.id: cgroup-brmrsm-productInventory
       configuration:
         max-retries: 3
         retry-delay-interval-ms: 1000
         record-fetch-timeout-ms: 1000
         brm-connection-retry-interval-ms: 30000
         strict-ordering: false
         buffer-strategy:
       topic-event-config:
         source-topic: productInventory
         sink-topic: productInventoryRetry
         key-state-topic: keyStateTopic
         consumer-thread-count: 1
         worker-thread-count: 2
         downstream-consumers:
           - connection-profile-id: "cp1"
             properties:
               group.id: cgroup-brmrsm-productInventory
             configuration:
               max-retries: 3
               retry-delay-interval-ms: 1000
               record.fetch.timeout.ms: 1000
               brm.connection.retry.interval.ms: 30000
             topic-event-config:
               source-topic: productInventoryRetry
               sink-topic: productInventoryDLQ
               consumer-thread-count: 1
               worker-thread-count: 2
               events:
                 - ProductCreateEvent
                 - ProductAttributeValueChangeEvent
                 - ProductStateChangeEvent
                 - ProductDeleteEvent
         events:
           - ProductCreateEvent
           - ProductAttributeValueChangeEvent
           - ProductStateChangeEvent
           - ProductDeleteEvent
     - connection-profile-id: "cp1"
       properties:
         group.id: cgroup-brmrsm-partyAccountInventory
       configuration:
```

```
    max-retries: 1
    retry-delay-interval-ms: 1000
    record-fetch-timeout-ms: 1000
    brm-connection-retry-interval-ms: 30000
    strict-ordering: false
    buffer-strategy:
  topic-event-config:
    source-topic: partyAccountInventory
    sink-topic: partyAccountInventoryRetry
    key-state-topic: keyStateTopic
    consumer-thread-count: 1
    worker-thread-count: 2
    downstream-consumers:
      - connection-profile-id: "cp1"
        properties:
          group.id: cgroup-brmrsm-partyAccountInventory
        configuration:
          max-retries: 1
          retry-delay-interval-ms: 1000
          record-fetch-timeout-ms: 1000
          brm-connection-retry-interval-ms: 30000
        topic-event-config:
          source-topic: partyAccountInventoryRetry
          sink-topic: partyAccountInventoryDLQ
          key-state-topic:
          consumer-thread-count: 1
          worker-thread-count: 2
          events:
            - PartyAccountCreateEvent
            - PartyAccountAttributeValueChangeEvent
            - PartyAccountStateChangeEvent
            - PartyAccountDeleteEvent
    events:
      - PartyAccountCreateEvent
      - PartyAccountAttributeValueChangeEvent
      - PartyAccountStateChangeEvent
      - PartyAccountDeleteEvent
- connection-profile-id: "cp1"
  properties:
    group.id: cgroup-brmrsm-billingAccountInventory
  configuration:
    max-retries: 1
    retry-delay-interval-ms: 1000
    record-fetch-timeout-ms: 1000
    brm-connection-retry-interval-ms: 30000
    strict-ordering: false
    buffer-strategy:
  topic-event-config:
    source-topic: billingAccountInventory
    sink-topic: billingAccountInventoryRetry
    key-state-topic: keyStateTopic
    consumer-thread-count: 1
    worker-thread-count: 2
    downstream-consumers:
      - connection-profile-id: "cp1"
        properties:
          group.id: cgroup-brmrsm-billingAccountInventory
        configuration:
          max-retries: 1
          retry-delay-interval-ms: 1000
          record-fetch-timeout-ms: 1000
          brm-connection-retry-interval-ms: 30000
```

```
            topic-event-config:
              source-topic: billingAccountInventoryRetry
              sink-topic: billingAccountInventoryDLQ
              key-state-topic:
              consumer-thread-count: 1
              worker-thread-count: 2
              events:
                - BillingAccountCreateEvent
                - BillingAccountAttributeValueChangeEvent
                - BillingAccountStateChangeEvent
                - BillingAccountDeleteEvent
        events:
          - BillingAccountCreateEvent
          - BillingAccountAttributeValueChangeEvent
          - BillingAccountStateChangeEvent
          - BillingAccountDeleteEvent
```

where,

- **connection-profile-id** is the identifier referencing the Kafka connection profile used by the consumer.
- **group.id** is the unique consumer group identifier for coordinating message consumption and offset management.
- **max-retries** is the maximum number of retry attempts for processing an event before routing it to the retry or DLQ topic.
- **retry-delay-interval-ms** is the delay in milliseconds between retry attempts after a processing failure.
- **record-fetch-timeout-ms** is the maximum time in milliseconds the consumer waits for records during a fetch operation. (Also appears as **record.fetch.timeout.ms**.)
- **brm-connection-retry-interval-ms** is the interval in milliseconds to wait between attempts to reconnect to BRM services. (Also appears as **brm.connection.retry.interval.ms**.)
- **strict-ordering** is a flag indicating whether strict ordering for message keys is enforced (true or false).
- **buffer-strategy** is the method used for buffering events within the consumer; implementation-specific.
- **source-topic** is the Kafka topic from which the consumer reads events.
- **sink-topic** is the Kafka topic to which failed events are sent for retries.
- **key-state-topic** is an optional topic used to track and manage state related to message keys.
- **consumer-thread-count** is the number of threads allocated for polling messages from the source topic.
- **worker-thread-count** is the number of threads used for processing polled events.
- **downstream-consumers** is a list of additional consumer stages for retry or DLQ handling, each with its configuration.
- **events** is the list of event types that the consumer instance is configured to process (e.g., ProductCreateEvent, PartyAccountDeleteEvent).

> **Note**
>
> It is recommended to use different **group.id** for every individual consumer. However consumers in the same downstream consumer chain can have the same **group.id**.

Example block for kafka consumers in the *REST_home***/scripts/application.yaml**

```
- connection-profile-id: "cp2"
  properties:
    username: ""
    password-key: "CP2_SCHEMA_WALLET_KEY"
    queue-name: OPEN_API_QUEUE
  wallet-location: "home/db_wallet"
  configuration:
    worker-thread-count: 1
```

where,

- **connection-profile-id** is the identifier referencing the database connection profile to be used by this configuration.
- **username** is the login name used for authenticating the connection to the database or queue.
- **password-key** is the key used to securely retrieve the authentication password (such as from a Kubernetes Secret or Wallet).
- **queue-name** is the name of the queue from which this consumer will read messages (e.g., for queue-based processing).
- **wallet-location** is the name of the wallet or credential store where sensitive authentication information is managed.
- **worker-thread-count** is the number of threads assigned for concurrent processing of messages from the queue.

Setting up Producers

> **Note**
>
> Configure this to recieve the TMF events for supported API requests and supported TMF consumer events.

Example block for producers in the *REST_home***/scripts/application.yaml**

```
producer:
    connection-profile-id: "cp1"
    properties:
    event-topics-config:
      - event-name: "ProductCreateEvent"
        topic-names: [ "productCreateTopic", "productCreateTopic2" ]
      - event-name: "ProductDeleteEvent"
        topic-names: [ "ProductDeleteTopic" ]
      - event-name: "ProductAttributeValueChangeEvent"
        topic-names: [ "productAttributeValueTopic" ]
      - event-name: "ProductStateChangeEvent"
        topic-names: [ "productStateChangeTopic" ]
      - event-name: "PartyAccountCreateEvent"
```

```
          topic-names: [ "partyAccountNotificationTopic" ]
        - event-name: "PartyAccountAttributeValueChangeEvent"
          topic-names: [ "partyAccountNotificationTopic" ]
        - event-name: "PartyAccountStateChangeEvent"
          topic-names: [ "partyAccountNotificationTopic" ]
        - event-name: "PartyAccountDeleteEvent"
          topic-names: [ "partyAccountNotificationTopic" ]
        - event-name: "BillingAccountCreateEvent"
          topic-names: [ "partyAccountNotificationTopic" ]
        - event-name: "BillingAccountAttributeValueChangeEvent"
          topic-names: [ "partyAccountNotificationTopic" ]
        - event-name: "BillingAccountStateChangeEvent"
          topic-names: [ "partyAccountNotificationTopic" ]
        - event-name: "BillingAccountDeleteEvent"
          topic-names: [ "partyAccountNotificationTopic" ]
```

where,

- **connection-profile-id** is the reference to the **id** of a defined connection profile (such as "cp1") that the producer will use for publishing events.
- **properties** is a collection of Kafka producer configuration parameters (such as acks, retries, batch size, etc.), set to control behavior of the producer and optimize performance, reliability, and security. These properties will override the **connection-profile** properties if they are configured.
- **event-topics-config** is a list that maps each event name to one or more Kafka topics to which the event will be published.
- **event-name** is the identifier for a specific business event type that will be produced and routed by this producer (e.g., "ProductCreateEvent").
- **topic-names** is the list of one or more Kafka topics to which messages for the corresponding **event-name** are published.

Completing Post-Installation Tasks

After deployment, validate the service using logs and functional checks. Customize event handling with mapper overrides if required and follow the correct procedure for restarting the REST Services Manager Notification Service after any configuration change.

To validate deployment and check logs:

1. Inspect the log directory for startup messages.
2. Confirm the service connects to Kafka brokers and joins the expected consumer group(s).
3. Verify correct topic subscriptions and successful offset commits.

To customize using **customMapperDirectory**:

- To override default event mappers, place custom mapping files in the **customMapperDirectory** defined in **application.yaml**. See "Configuring and Adding Custom Mapper Files" for more information.
- Restart the REST Services Manager Notification Service after adding or modifying custom mapper files to apply changes.

To restart REST Services Manager:

- After any configuration or credential change, restart the REST Services Manager Notification Service to load the updated settings and wallet entries.

- Use the supplied service management commands or scripts for safe restart procedures. Stop the BRM REST Services Manager by running the *REST_home***/scripts/stop-brm-rsm.sh** script. Start the BRM REST Services Manager by running the *REST_home***/scripts/start-brm-rsm.sh** script.

Performing Verification and Sanity Checks

Conduct a series of checks to ensure all topics, partitions, security settings, and service connections are correct. This section includes procedures for performing functional connectivity testing and verifying end-to-end message flow.

Perform checks to validate the REST Services Manager Notification Service deployment:

- Confirm all topics, partitions, and ACLs are present in Kafka.
- Ensure consumers join the correct consumer group and receive partition assignments.
- Check that the service successfully establishes SSL or SASL connections to brokers.
- Run test messages through the pipeline to validate end-to-end message flow.
- Use built-in scripts to verify wallet entry references and keystore integrity.

Security Best Practices for REST Services Manager Notification Service

Adhere to recommended security practices for storage and handling of credentials. This section summarizes wallet usage, credential rotation, file permissions, and compliance alignment.

To maintain a secure deployment:

- Store all credentials, passphrases, and secrets in the Oracle wallet. Never use plain text or version control for secret values.
- Limit file permissions on wallet and keystore files to only necessary service accounts.
- Rotate credentials and certificates periodically, updating the wallet, and restart services after changes.
- Do not disable SSL endpoint identification unless explicitly required and with a formal risk assessment.
- Ensure your deployment always aligns with Oracle security and compliance policies.

Troubleshooting SSL Connection Errors

BRM REST Services Manager can use SSL connections for invoking OAuth Token Introspection Endpoint and Oracle Unified Directory REST APIs. Table 11-6 describes the possible SSL connection errors and solutions.

Table 11-6 SSL Connection Errors and Solutions

SSL Connection Errors	Solutions
javax.net.ssl.SSLHandshakeException: java.security.cert.CertificateException: No name matching hostname found	Ensure that the Common Name (CN) in the SSL certificate is the fully qualified domain name of the server where endpoints are hosted.

Table 11-6 (Cont.) SSL Connection Errors and Solutions

SSL Connection Errors	Solutions
javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilder Exception: unable to find valid certification path to requested target	Ensure that the SSL certificate of the endpoint is imported into the Java KeyStore.

12

Troubleshooting the BRM Installation

Learn how to solve problems that may occur in your Oracle Communications Billing and Revenue Management (BRM) system.

Topics in this document:

- Problems Installing the Database
- Problems Installing BRM
- Problems Connecting to the Database

For more information, see "Reference Guide to BRM Error Codes" and "Resolving Problems in Your BRM System" in *BRM System Administrator's Guide*.

Problems Installing the Database

For more information on errors that can occur when installing your database, see your database documentation. For specific issues, see the following sections:

- Problem: Cannot Start the Oracle Listener
- Problem: Cannot Connect to the Oracle Database as the System User

Problem: Cannot Start the Oracle Listener

When you try to start the Oracle Listener, you get a “permission denied" error.

Possible Cause

The path points to a directory for which you do not have access.

Solution

Change the permissions to the directory where the **listener.ora** and **listener.log** files are stored.

Problem: Cannot Connect to the Oracle Database as the System User

You cannot connect to the database as the **system** user with SQL.

Possible Causes

- The Oracle Listener is not running.
- Oracle cannot find a **tnsnames.ora** file.
- The **tnsnames.ora** file does not point to a valid database.

Solution

1. Start the Oracle Listener.
2. Make sure the **tnsnames.ora** file points to your database on the specific host machine.
3. Make sure the Oracle user's path contains the correct path to your **tnsnames.ora** file.

Problems Installing BRM

If you have a problem installing the BRM software or its optional components, first check the **pin_setup.log** file, and then the Data Manager (DM) and Connection Manager (CM) **log** files for possible reasons for the failure. For specific issues, see the following sections:

- Problem: An Error Occurred When Selecting an Optional Component to Install
- Problem: An Error Occurred When pin_setup is Run During Installation
- Problem: An Error Occurred during the Move Data Process
- Problem: Errors Are Recorded in the Log File When dm_oracle Starts
- Problem: An Error Occurs in the dm_fusa.pinlog file during pin_setup of Paymentech Manager
- Problem: An Error is Recorded in the dm_oracle.pinlog and dm_oracle Fails to Start

Problem: An Error Occurred When Selecting an Optional Component to Install

When you install one or more new optional components, either during or after an upgrade from BRM 15.0.x, you may receive an error message like the following:

```
The distribution Billing Revenue Management 15.2.0.0.0 contains incompatible features
with the following:
Billing Revenue Management 15.0.1.0.0 [CORE 15.2.0.0.0->CORE 15.0.1.0.0 (CORE 15.2.0.0.0-
>[CORE 15.0.1.0.0]))]
```

The error message may contain more or different incompatible features. It is still the same situation and fix.

Possible Cause

There is a conflict in the **registry.xml** file.

Solution

To remove this error:

1. Take a note or screenshot of the complete errors received.
2. Stop the installer.
3. Open the *BRM_home***/inventory/registry.xml** file in a text editor.
4. Find the entry in the file that corresponds to the specifics of the first error received. You are looking for the <feature> element with a name attribute that matches what is in the error. For example, for this error:

```
The distribution Billing Revenue Management 15.2.0.0.0 contains incompatible
features with the following:
Billing Revenue Management 15.0.1.0.0 [CORE 15.2.0.0.0->CORE 15.0.1.0.0 (CORE
15.2.0.0.0->[CORE 15.0.1.0.0])]
```

look for the following element:

```
<feature status="installed" name="CORE" version="15.0.1.0.0">
   <sessions>
      <session id="1" date="2024-10-22T09:30:07.000-07:00" action="install"/>
   </sessions>
   <components>
      <component status="installed" name="CORE" version="15.0.1.0.0">
         <sessions>
            <session id="1" date="2024-10-22T09:30:07.000-07:00" action="install"/>
         </sessions>
         <targets>
            <target qualifier="filegroup1.jar+" source="filegroup1.jar"
symbol="root.symbol" location="" status="installed">
               <sessions>
                  <session id="1" date="2024-10-22T09:30:07.000-07:00"
action="install"/>
               </sessions>
            </target>
            <target qualifier="filegroup2.jar+oui" source="filegroup2.jar"
symbol="oracle.nginst.core.symbol" location="oui" status="installed">
               <sessions>
                  <session id="1" date="2024-10-22T09:30:07.000-07:00"
action="install"/>
               </sessions>
            </target>
         </targets>
      </component>
   </components>
</feature>
```

5. Delete the entire matching element.
6. Repeat for any additional errors seen in the installer.
7. Save and close the **registry.xml** file and run the installer or upgrade again. The error should not be present.

Problem: An Error Occurred When pin_setup is Run During Installation

When you run the **pin_setup** script of BRM during installation, error occurs if the database alias is incorrect.

Possible Cause

The database alias is incorrect, so the **pin_setup** script cannot finish executing.

Solution

In single-schema environments, make sure you do the following and then continue the BRM installation:

1. Set the $TNS_SERVICE_NAME environment variable to point to the database service name.
2. Open the **tnsnames.ora** file in a text editor.

By default, that file is in the *Oracle_home***/network/admin/** directory.

3. Set SERVICE_NAME as the database alias in the connect descriptor:

   For example:

   ```
   pindb.example.com =(DESCRIPTION =(ADDRESS_LIST =(ADDRESS = (PROTOCOL = ) (HOST = )
   (PORT = ) ))(CONNECT_DATA =(SERVICE_NAME = pindb.example.com)))
   ```

4. Save and close the file.

In multischema environments, make sure you do the following after installing BRM:

1. Specify the correct database alias in the Oracle wallet.

   For updating the configuration entries in the Oracle wallet, see "About Oracle Wallet" in *BRM System Administrator's Guide*.

2. Update the **$MAIN_DB{'alias'}** entry in the *BRM_home***/setup/pin_setup.values** file to point to the correct database alias.

3. Run the **pin_setup** script. See "Running the pin_setup Script".

Problem: An Error Occurred during the Move Data Process

You receive the error “An error occurred during the move data process -132."

Possible Cause

A BRM process is running, so the **pin_setup** script cannot finish executing.

Solution

Make sure you stop all existing BRM processes, and then run the **pin_setup** script again.

Problem: Errors Are Recorded in the Log File When dm_oracle Starts

After you install BRM and while running the **pin_setup** script of BRM, when the Oracle DM starts, errors are recorded in the **dm_oracle.pinlog** file.

Possible Cause

Cannot find the ACCOUNT_T table, as this gets created later on.

Solution

There is no solution; the error does not impact any functionality and can be ignored.

Problem: An Error Occurs in the dm_fusa.pinlog file during pin_setup of Paymentech Manager

Errors occur while running the **pin_setup** script of Paymentech Manager in the **dm_fusa.pinlog** file.

Possible Cause

The simulator is not started.

Solution

After the **pin_setup** of Paymentech Manager is complete, start the answer simulator to remove the errors. The simulators are in *BRM_home***/bin**. You start the simulator through the command line:

```
start_answer
```

Problem: An Error is Recorded in the dm_oracle.pinlog and dm_oracle Fails to Start

After you install BRM 15.x on a server running Red Hat Enterprise Linux (RHEL) 8.7 for the first time, **dm_oracle** fails to start and an error is recorded in the **dm_oracle.pinlog** file like the following:

```
M Thu Oct 17 02:49:59 2024 <HOSTNAME> dm:1585533 dm_main.c(82):3009
3:<HOSTNAME>:dm:1585508:45774656:0:1729158599680687:0::::-1
DM(back end [0] - 1585533): ERROR bad dlopen for: "/opt/portal/15.0/BRM/sys/
dm_oracle/dm_oracle19c.so": libnsl.so.1: cannot open shared object file: No
such file or directory

E Thu Oct 17 02:49:59 2024 <HOSTNAME> dm:1585508 dm_main.c(82):2779
3:<HOSTNAME>:dm:1585508:45774656:0:1729158599680687:0::::-1
DMm be #0 init failed, exiting
```

Possible Cause

The **libnsl.so.1** file is not installed on the server.

Solution

You must install the network services library package to resolve this error.

To install the network services library package:

1. Log in as the **root** user on the server.
2. Install the required package using one of the following commands:

   ```
   yum install libnsl.x86_64
   ```

   ```
   dnf install libnsl.x86_64
   ```

3. Ensure that a new **libnsl*.so** file is installed in the **/lib64** directory.
4. Ensure that a symbolic link named **libnsl.so.1** is created and points to the newly installed **libnsl*.so** file.

Problems Connecting to the Database

Conflicting settings in the **pin_setup.values** configuration file often cause database connection failures. For specific issues, see the following sections:

- Problem: Pointer to the Wrong Database Name

- Problem: Cannot Log In to the Database
- Problem: Out of Memory on Linux

Problem: Pointer to the Wrong Database Name

The **pin_setup** script cannot connect to the database, and the error message in the DM log file shows that the DM is trying to connect to the incorrect database.

Possible Cause

The **$MAIN_DB{'alias'}** entry in the **pin_setup.values** file does not match the database alias.

Solution

Enter the correct name for your database; for example, **pindb***hostname*, in the **$MAIN_DB{'alias'}** entry of the **pin_setup.values** file. Re-run the **pin_setup** script.

Problem: Cannot Log In to the Database

The **pin_setup** script cannot log in to the BRM database. The error message indicates an invalid user name or password.

Possible Cause

The **$MAIN_DB{'user'}** and **$MAIN_DB{'password'}** entries in the **pin_setup.values** file do not specify the correct user name and password for the BRM database.

Solution

First, verify that you can connect to the BRM database by using the user name and password you want. If you cannot connect:

1. Use SQL statements to set up the user and password for the BRM database.
2. Enter the correct user name and password for the BRM database in the **$MAIN_DB{'user'}** and **$MAIN_DB{'password'}** entries in the **pin_setup.values** file.
3. Re-run the **pin_setup** script.

Problem: Out of Memory on Linux

The **pin_setup** script cannot start the DM, and the error log file for the DM refers to “bad shmget", “bad shmat", or another error related to memory.

Possible Cause

There is not enough shared memory on the database server.

Solution

Increase the shared memory value in the **/etc/sysctl.conf** file. Perform these steps in a text editor:

1. Log on as root.

2. Open the **/etc/sysctl.conf** file.
3. Add the following lines to the end of the file, or modify the values if these lines are already present:

   ```
   kernel.shmmax = 536870912
   kernel.shmall = 536870912
   ```

4. Save and close the file.
5. Run **sysctl -p** from the Linux prompt.

13

Installing Multiple BRM Instances on One System for Testing

Learn how to help your developers test their system by installing and setting up multiple Oracle Communications Billing and Revenue Management (BRM) instances.

Topics in this document:

- Installing and Configuring Multiple Instances of BRM on One Machine
- Reinitializing the Database
- Removing BRM Data from the Database

Installing and Configuring Multiple Instances of BRM on One Machine

You can install multiple instances of BRM on one machine to reduce the amount of hardware needed to test the BRM software. To set up multiple instances of BRM on one machine, perform the following steps:

1. Creating Multiple Users
2. Installing and Configuring the Oracle Database for Multiple Instances
3. Installing Multiple Instances of BRM
4. Configuring Each BRM Instance

Creating Multiple Users

Create a user for each instance of BRM you want to install. For example, to install four instances of BRM, create users **pin1**, **pin2**, **pin3**, and **pin4**.

To create multiple users:

1. Log in as **root**:

   ```
   % su - root
   ```

2. Create a group. For example, run the following command to create the group **pin**:

   ```
   # groupadd -g 9500 pin
   ```

3. Create each user. For example, run the following commands to create four users:

   ```
   # useradd -g pin -s /bin/csh -d BRM_home/pin1 pin1
   # useradd -g pin -s /bin/csh -d BRM_home/pin2 pin2
   # useradd -g pin -s /bin/csh -d BRM_home/pin3 pin3
   # useradd -g pin -s /bin/csh -d BRM_home/pin4 pin4
   ```

4. Create a password for each user.

Installing and Configuring the Oracle Database for Multiple Instances

Before you install BRM, you must install the Oracle database and configure it for multiple instances of BRM. Install and configure the Oracle database according to the instructions in "BRM Preinstallation Tasks", except create your BRM tablespaces and BRM users as described in this section. You can also create the BRM tablespaces and BRM users during the BRM server installation.

(Optional) Creating Tablespaces for Each User

You can create unique data, index, and temporary tablespaces for each instance of BRM you install. The examples in this document use the following tablespaces:

- **pin*N*00** (for data)
- **pinx*N*00** (for indexes)
- **PINTEMP*N*** (for a temporary tablespace)

1. Create a directory for the tablespaces, such as **/u02/oradata/pindb**.

   This directory is referred to as *table_location*.

2. Connect to the Oracle database by using SQL*Plus:

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

3. Create a data, index, and temporary tablespace for each BRM user.

   Repeat the following commands for each BRM instance:

   ```
   SQL> create tablespace pin00 datafile 'table_location/pin00.dbf'
        size 600M reuse autoextend on extent management
        local uniform size 64K segment space management
        auto;

   Tablespace created.

   SQL> create tablespace pinx00 datafile 'table_location/pinx00.dbf'
        size 400M reuse autoextend on extent management
        local uniform size 64K segment space management
        auto;

   Tablespace created.

   SQL> create temporary tablespace pintemp tempfile
        'table_location/PINTEMP.dbf' size 100M reuse autoextend on;

   Tablespace created.
   ```

(Optional) Creating Multiple BRM Users

Create the BRM users who can access the Oracle database. The examples in this document use **pin*N***, but you can use another naming scheme.

1. Connect to the Oracle database by using SQL*Plus:

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

2. Perform the following for each instance of BRM you will install:

   ```
   SQL> create user pinN identified by password;

   User created.

   SQL> grant resource, connect to pinN;

   Grant succeeded.

   SQL> alter user pin default tablespace pinN00;

   User altered.

   SQL> alter tablespace PINTEMP temporary;

   Tablespace altered.

   SQL> alter user pin temporary tablespace PINTEMP;

   User altered.
   ```

3. Type **exit** to exit SQL*Plus.

Installing Multiple Instances of BRM

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

To install multiple instances of BRM on one machine:

1. Install the PERL libraries and the JRE required to install BRM components.
2. Log in as user **pin***N*.

   ```
   % su - pinN
   ```

3. Go to the directory in which you installed the PERL libraries and JRE, and then source the **source.me** file.

   > **Caution**
   >
   > You must source the **source.me** file to proceed with installation. Otherwise, “suitable JVM not found" and other error messages appear.

   Bash shell:

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

   C shell:

   ```
   source source.me.csh
   ```

4. Stop all BRM processes. See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

5. Install BRM in each instance. See "Installing BRM".

   > **Note**
   >
   > You must log in separately for each instance of BRM that you need to install. For example, log in as **pin1** and install the first instance of BRM. Then, log in as **pin2** and install the second instance of BRM, and so on.

6. Go to the directory in which you installed the BRM software, and then source the **source.me** file:

   Bash shell:

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

   C shell:

   ```
   source source.me.csh
   ```

7. To further configure BRM, such as by changing the default currency and country, edit the *BRM_home***/setup/pin_setup.values** file.

   This file stores the information you provided to the installer and a number of database and add-on component parameters.

Each instance of BRM is installed in the *BRM_home***/pin***N* directory, where *N* represents the instance number. For example, the installation program copies BRM to the following directories when you install four instances of BRM in the default BRM directory, **/opt/portal/15.2.0.0.0**:

```
/opt/portal/15.2.0.0.0/pin1
/opt/portal/15.2.0.0.0/pin2
/opt/portal/15.2.0.0.0/pin3
/opt/portal/15.2.0.0.0/pin4
```

Configuring Each BRM Instance

Follow these steps for each instance of BRM you want to install on your machine:

1. Log in as **root** and change the permissions for the *BRM_home***/pin***N* directory and its contents from **pin** to **pin***N*:

   ```
   % su - root
   # cd BRM_home
   # chown -R pin pinN
   ```

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

   ```
   % cd BRM_home/setup
   % ./pin_setup
   ```

3. Check the **pin_setup.log** file for status and errors.
4. Verify that BRM was installed and set up correctly by creating an account with Billing Care.
5. Repeat steps 1 through 4 for each instance of BRM you installed on your machine.

Reinitializing the Database

You can reinitialize the BRM database, including tables, indexes, and triggers, and return it to its original condition. You might do this, for example, to clean out a test database.

> **Note**
>
> When you reinitialize a database, you completely remove existing data from the database. When the data is removed, you cannot restore it. Perform this task only if you are certain you will not need to access any data in the database or you have backed up the data and have confirmed that you can restore it.

Reinitializing BRM Server and Optional Component Data

To reinitialize all BRM data, including data in your core BRM tables and optional component tables, do the following:

1. Stop all BRM processes. See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.
2. Edit the **pin_setup.values** file.
   1. Log in as **pin**, go to the *BRM_home***/setup** directory, and open the **pin_setup.values** file in a text editor such as **vi**:

      ```
      % su - pin
      % cd BRM_home/setup
      % vi pin_setup.values
      ```

   2. Change the **$SETUP_DROP_ALL_TABLES** and **$SETUP_INIT_DB** entries to **YES**:

      ```
      $SETUP_DROP_ALL_TABLES = "YES";
      $SETUP_INIT_DB = "YES";
      ```

   3. Save and close the file.
3. Run the **pin_setup** script:

   ```
   % cd BRM_home/setup
   % ./pin_setup
   ```

4. Drop the optional component tables from your database.
5. Using SQL*Plus, log in to your database as the SYSTEM user and run the following command using the information in Table 13-1:

   ```
   SQL> @path/file_name
   ```

Table 13-1 Optional Component Drop Source File Location and Name

Optional Component	Path	File Name
GPRS Manager	*BRM_home***/sys/dd/data**	**drop_tables_gprs_oracle.source**
GSM Manager	*BRM_home***/sys/dd/data**	**drop_tables_service_order_oracle.source** **drop_tables_settlement_oracle.source**
Invoice DM	*BRM_home***/sys/dm_invoice/data**	**drop_tables.source**
Number Manager	*BRM_home***/sys/dd/data**	**drop_tables_num_oracle.source**
Rated Event Loader	*BRM_home***/sys/dd/data**	**drop_tables_rel_oracle.source**
SIM Manager	*BRM_home***/sys/dd/data**	**drop_tables_sim_oracle.source**
Vertex Manager	*BRM_home***/sys/dd/data**	**drop_tables_telephony_oracle.source**
Suspense Manager	*BRM_home***/sys/dd/data**	**drop_tables_suspense_oracle.source**

Table 13-1 (Cont.) Optional Component Drop Source File Location and Name

Optional Component	Path	File Name
Services Framework Manager	*BRM_home*/**sys/dd/data**	**drop_tables_telco_oracle.source** **drop_tables_config_accountera_oracle.source**
IP Address Manager	*BRM_home*/**sys/dd/data**	**drop_tables_ip_oracle.source** **drop_tables_apn_oracle.source**

6. Start all BRM processes.

   See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

All core BRM tables and optional component tables have been dropped from your system. You can start adding test accounts to your database and continue testing.

Reinitializing Optional Component Data Only

To reinitialize data from your optional component tables, but keep the data in your core BRM tables:

1. Stop the CM and DM processes.
2. Enable writing of the data dictionary objects in your Oracle DM configuration file:
   a. Open your Oracle DM configuration file (*BRM_home*/**sys/dm_oracle/pin.conf**).
   b. Make sure the following entries are set to **1**:

   ```
   - dm dd_write_enable_objects  1
   - dm dd_write_enable_fields  1
   - dm dd_write_enable_portal_objects  1
   ```

   c. Save and close the file.
3. Stop and restart the CM and Oracle DM processes.
4. Delete the BRM data dictionary objects for each optional component.

   You can find the data dictionary objects for each component in the files listed in Table 13-2.

Table 13-2 Optional Component Data Dictionary Drop Source File Location and Name

Optional Component	Path and File Name
GPRS Manager	*BRM_home*/**sys/dd/data/dd_objects_gprs.source**
GSM Manager	*BRM_home*/**sys/dd/data/dd_objects_service_order.source** *BRM_home*/**sys/dd/data/dd_objects_telco_gsm.source** *BRM_home*/**sys/dd/data/dd_objects_settlement.source** *BRM_home*/**sys/dd/data/dd_objects_telco.source** *BRM_home*/**sys/dd/data/dd_objects_config_accountera.source**
Invoice DM	*BRM_home*/**sys/dm_invoice/data/dd_objects.source**
Number Manager	*BRM_home*/**sys/dd/data/dd_objects_num.source**
Rated Event Loader	*BRM_home*/**sys/dd/data/dd_objects_rel.source**
SIM Manager	*BRM_home*/**sys/dd/data/dd_objects_sim.source**
Vertex Manager	*BRM_home*/**sys/dd/data/dd_objects_telephony.source**
Suspense Manager	*BRM_home*/**sys/dd/data/dd_objects_suspense.source**

Table 13-2 (Cont.) Optional Component Data Dictionary Drop Source File Location and Name

Optional Component	Path and File Name
IP Address Manager	*BRM_home*/**sys/dd/data/dd_objects_ip.source** *BRM_home*/**sys/dd/data/dd_objects_apn.source**

a. Create an flist that contains the POID of the objects you want to delete.

   See "Understanding flists and Storable Classes" in *BRM Developer's Guide*.

b. Run the PCM_OP_SDK_DEL_OBJ_SPECS opcode with the input flist you created. For more information, see "Deleting Storable Class Specifications" in *BRM Developer's Guide*.

5. Stop the CM and Oracle DM processes.

   See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

6. Drop the optional component tables from your database.

7. Using SQL*Plus, log in to your database as the SYSTEM user and run the following command using the values in Table 13-3:

```
SQL> @path/file_name
```

Table 13-3 Optional Component Drop Source File Location and Name

Optional Component	Path	File Name
GPRS Manager	*BRM_home*/**sys/dd/data**	**drop_tables_gprs_oracle.source**
GSM Manager	*BRM_home*/**sys/dd/data**	**drop_tables_service_order_oracle.source** **drop_tables_settlement_oracle.source**
Invoice DM	*BRM_home*/**sys/dm_invoice/data**	**drop_tables.source**
Number Manager	*BRM_home*/**sys/dd/data**	**drop_tables_num_oracle.source**
Rated Event Loader	*BRM_home*/**sys/dd/data**	**drop_tables_rel_oracle.source**
SIM Manager	*BRM_home*/**sys/dd/data**	**drop_tables_sim_oracle.source**
Vertex Manager	*BRM_home*/**sys/dd/data**	**drop_tables_telephony_oracle.source**
Suspense Manager	*BRM_home*/**sys/dd/data**	**drop_tables_suspense_oracle.source**
Services Framework Manager	*BRM_home*/**sys/dd/data**	**drop_tables_telco_oracle.source** **drop_tables_config_accountera_oracle.source**
IP Address Manager	*BRM_home*/**sys/dd/data**	**drop_tables_ip_oracle.source** **drop_tables_apn_oracle.source**

8. Restore the entries in your Oracle DM configuration file to their original value.

   a. Open your Oracle DM configuration file (*BRM_home*/**sys/dm_oracle/pin.conf**).

   b. Return the following entries to their original value:

```
- dm dd_write_enable_objects  0
- dm dd_write_enable_fields  0
- dm dd_write_enable_portal_objects  0
```

   c. Save and close the file.

9. Stop and restart the CM and Oracle DM processes.

All optional component tables have been dropped from your system. You can start adding test accounts to your database and continue testing.

Removing BRM Data from the Database

To completely remove BRM data, including tables, indexes, and triggers, from the Oracle database:

> **Note**
>
> After you remove the data, you cannot restore it. Perform this task only if you are certain you will not need to access any data in the database or if you have backed up the data and have confirmed that you can restore it.

1. Use SQL to connect to the database as the **system** user:

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

2. Stop all BRM processes.

   See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

3. Enter the following command, replacing *pin_user* with the user you created for BRM, such as **pin**:

   ```
   SQL> DROP USER pin_user CASCADE;
   ```

4. Log in to the database server as *pin_user*.

   If you receive an error indicating an invalid user name, the operation was successful.

14

Uninstalling BRM

Learn how to uninstall the Oracle Communications Billing and Revenue Management (BRM) software, its server components, and its client applications from your computer.

For details on how to uninstall or reinstall the Oracle database, see the database vendor's documentation for your platform.

Topics in this document:

- Uninstalling the BRM Software
- Uninstalling BRM in Silent Mode

Uninstalling the BRM Software

You can use Oracle Universal Installer to uninstall the BRM software.

Uninstalling the BRM software requires three tasks:

1. Uninstalling Optional Components
2. Uninstalling BRM
3. Uninstalling BRM Client Applications

For uninstalling BRM server or client applications in silent mode, see "Uninstalling BRM in Silent Mode".

> ⓘ **Note**
>
> Uninstalling BRM does not uninstall the configuration files generated during the BRM installation. You must remove these files manually.

Uninstalling Optional Components

To uninstall optional components:

> ⓘ **Note**
>
> - Optional components must be uninstalled before uninstalling BRM.
> - If you are uninstalling Oracle Analytics Publisher reports, you must uninstall all of the Oracle Analytics Publisher reports installed. You cannot uninstall individual reports in the BRM-Oracle Analytics Publisher invoicing integration package.

1. Back up the BRM database.

   See "Maintaining a BRM Database".

2. Stop all BRM daemons, processes, and managers.

   See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

3. Log in as user **pin**.
4. Go to the *BRM_home***/oui/bin** directory, where *BRM_home* is the directory in which you installed BRM.
5. Run the following command from a GUI (for example, VNC Server):

   ```
   ./deinstall.sh
   ```

   If multiple products are installed in the same location, you can run the following command:

   ```
   ./deinstall.sh -paramFile../../inventory/inis/
   distribution_name_distribution_version.ini
   ```

   where:

   - *distribution_name* is the name of the distribution for the optional components installed.
   - *distribution_version* is the version number of the distribution for the optional components installed.

   The Distribution to Uninstall window appears.

6. Select the components you want to uninstall.

   > **Note**
   >
   > You must uninstall the dependent components before uninstalling the required components. For example, you can uninstall EAI Manager only after you uninstall the dependent optional components, such as Account Synchronization Manager.

7. Click **Uninstall**.

   The Welcome window appears.

8. Click **Next**.

   The Uninstallation Summary window appears.

9. Click **Uninstall**.

   The Uninstallation Progress window appears.

10. Click **Next**.

    The Uninstallation Complete window appears.

11. Click **Finish**.

Uninstalling BRM

To uninstall BRM:

1. Back up the BRM database.

   See "Maintaining a BRM Database".

2. Stop all BRM daemons, processes, and managers. See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.
3. Log in as user **pin**.

4. Go to the *BRM_home***/oui/bin** directory, where *BRM_home* is the directory in which you installed BRM.
5. Run the following command:

   ```
   ./deinstall.sh
   ```

   The Distribution to Uninstall window appears.
6. Select the components you want to uninstall.

   > **Note**
   >
   > Dependent components must be uninstalled before uninstalling the required components. For example, you can uninstall EAI Manager only after you uninstall the dependent optional components, such as Account Synchronization Manager.

7. Click **Uninstall**.

   The Welcome window appears.
8. Click **Next**.

   The Uninstallation Summary window appears.
9. Click **Uninstall**.

   The Uninstallation Progress window appears.
10. Click **Next**.

    The Uninstallation Complete window appears.
11. Click **Finish**.

Uninstalling BRM Client Applications

This section describes how to uninstall BRM client applications on Linux and Windows.

Uninstalling BRM Client Applications on Linux

To uninstall BRM client applications on Linux:

1. Back up the BRM database.

   See "Maintaining a BRM Database".
2. Go to the *BRM_client_home***/oui/bin** directory, where *BRM_client_home* is the directory in which the BRM client applications are installed.
3. Run the following command:

   ```
   ./deinstall.sh
   ```

   The Distribution to Uninstall window appears.
4. Select the client applications you want to uninstall.
5. Click **Next**.

   The Welcome window appears.
6. Click **Next**.

The Uninstallation Summary window appears.

7. Click **Uninstall**.

   The Uninstallation Progress window appears.

8. Click **Next**.

   The Uninstallation Complete window appears.

9. Click **Finish**.

Uninstalling BRM Client Applications on Windows

To uninstall BRM client applications on Windows:

1. Ensure that 64-bit JDK is installed on your Windows system. It is required by the BRM OUI uninstaller.

   For the latest supported version of JDK, see "Additional BRM Software Requirements" in *BRM Compatibility Matrix*.

2. Back up the BRM database.

   See "Maintaining a BRM Database".

3. Do one of the following to start the uninstaller:

   - (Windows 8.1 and Windows 10) Use the Windows uninstall feature.
   - (Other Windows versions) Do one of the following:
     - Run the **deinstall.cmd** script from the *BRM_client_home***\oui\bin** directory.
     - Use the Windows uninstall feature.

   The Distribution to Uninstall window appears.

4. Select the client applications you want to uninstall.

5. Click **Uninstall**.

   The Welcome window appears.

6. Click **Next**.

   The Uninstallation Summary window appears.

7. Click **Uninstall**.

   The Uninstallation Progress window appears.

8. Click **Next**.

   The Uninstallation Complete window appears.

9. Click **Finish**.

Uninstalling BRM in Silent Mode

> **Note**
>
> You can uninstall individual BRM components only using GUI mode.

To uninstall BRM (server or client applications) in silent mode:

1. Go to the *BRM_home***/oui/bin** directory.
2. Run the following command:

   ```
   ./deinstall.sh -responseFile path -silent -uninstall
   ```

   where *path* is the absolute path to the response file that you created in the root directory during the GUI installation of BRM or the BRM client applications.

Part III

Upgrading BRM

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 to BRM 15.2
- Upgrading BRM 12.0 to BRM 15.2
- Upgrading BRM 15.x to BRM 15.2
- Rolling Back Your BRM Database Upgrade
- Troubleshooting the BRM Upgrade

15

About Upgrading to BRM 15.2

Learn about upgrading your existing system to the latest Oracle Communications Billing and Revenue Management (BRM) release.

Topics in this document:

- About Upgrading BRM
- 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 existing 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 15.1 to BRM 15.2, BRM 15.1 is the old release and BRM 15.2 is the new release.

> **Note**
>
> - A direct upgrade from the BRM 7.5 release is not supported. If you are upgrading from BRM 7.5, you must upgrade your system to BRM 12.0 first. See *Compatibility Matrix* for BRM 12.0 for information about supported BRM 7.5-to-BRM 12.0 upgrades.
> - For information about the BRM 12.0 releases from which you can directly upgrade to BRM 15.2, see "Supported BRM Upgrades" in *BRM Compatibility Matrix*. For releases where you cannot directly upgrade to the desired 15.2 release from your BRM 12.0 release, upgrade to a later 12.0 patch set and then upgrade from that patch set to BRM 15.2.

About Upgrading BRM

Upgrading BRM 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 15.2** (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.

> **Note**
>
> If you use a separate Invoice database schema, you must also upgrade that database schema. Apply any upgrade scripts applied to the main BRM database schema to the Invoice database schema as well.

- **Implement customizations for the new version**.
  - 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 15.2 (without installing the database schema).
6. Upgrade the BRM database schema.
7. Install BRM 15.2 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.

Considerations for Upgrading from BRM 12.0

There are some changes specific to upgrading from BRM 12.0 that might have some impact on your upgrade:

- 64-bit support:
  - All of the BRM libraries are built in a 64-bit environment. Because of this, the 64 suffix is removed from all of the libraries.

  - A new field, PIN_FLDT_INT64 has been added. This enables the storage of large numbers.
  - Consult *BRM Compatibility Matrix* to ensure that you are using the supported version s of compilers and libraries.
  - See "Migrating Custom Code to a 64-bit Environment" for information about the upgrade impacts from this change.
- CM Authorization:
  - Starting with BRM 15.0, BRM provides fine-grained authorization for pcm_client and thick client applications. This is implemented using the new storable class **/config/role** and the following new opcodes:
    - PCM_OP_CUST_CREATE_ROLE
    - PCM_OP_CUST_ASSOCIATE_ROLE
    - PCM_OP_CUST_UPDATE_ROLE
    - PCM_OP_CUST_DELETE_ROLE
  - You can configure whether to use this new authorization method by setting the new system business parameter, **CMAuthorization**. This parameter is enabled by default.
  - For more information about the default roles and the implementation of this feature, see *System Administrator's Guide*. For more information about the new opcodes, see *Opcode Guide*.
- Merging **dm_ifw_sync** and **dm_aq** with **dm_oracle**:
  - The **dm_ifw_sync** and **dm_aq** queues were previously used for Advanced Queuing in BRM. However, the Account Sync Manager (dm_ifw_sync) and the Queue Manager (dm_aq) components were deprecated in BRM 15.0. Their functionality has been moved to **dm_oracle**, which has been enhanced to queue these messages going forward.
  - See "Updating Queue Handling" for information about the upgrade impacts from this change.

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 the database, you might need to upgrade your database software before upgrading BRM. For more information, see the following:

  – See "System Requirements" and *BRM Compatibility Matrix* for a list of software supported by BRM.

  – See "Planning Your Database Configuration" for general database information.

- 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 "Improving BRM Performance" in *BRM System Administrator's Guide* for information on memory configuration.

- 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 Linux machine. See "Installing and Configuring Multiple Instances of BRM on One Machine" for more information.
>
> – 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 patches 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" 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)".

16
Upgrading BRM 12.0 to BRM 15.2

Learn about the procedures required to upgrade Oracle Communications Billing and Revenue Management (BRM).

Before performing this upgrade, see "About Upgrading to BRM 15.2" for information on how to plan, prepare for, and test your upgrade.

Topics in this document:

- About the Upgrade
- Tasks Involved in the Upgrade Process
- Preparing for the Upgrade
- Upgrading to BRM 15.2
- Upgrading Your Data to BRM 15.2
- Completing the Upgrade
- Post-Upgrade Procedures

> **Note**
>
> - A direct upgrade from the BRM 7.5 release is not supported. If you are upgrading from BRM 7.5, you must upgrade your system to BRM 12.0 first.
> - For information about the BRM 12.0 releases from which you can directly upgrade to BRM 15.2, see *BRM Compatibility Matrix*.

About the Upgrade

To upgrade from BRM 12.0 to BRM 15.2, upgrade the following components:

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

Tasks Involved in the Upgrade Process

This section provides a list of tasks required to upgrade from BRM 12.0 to BRM 15.2. Some tasks are optional or apply only to certain 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 12.0 data. See "Creating a Complete Backup of Your BRM 12.0 Data".
4. Create a new schema in BRM 15.2. See "Creating the Database for Your New Version".
5. Import your data BRM 12.0 data. See "Importing Your BRM 12.0 Data".
6. Install Java. See "Installing Java".
7. Install Perl. See "Installing Perl".
8. Obtain the JDBC driver. See "Downloading the JDBC Driver".
9. Set the environment variables. See "Setting the Environment Variables for Upgrade".
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. Point your BRM 15.2 installation to the new database schema. See "Pointing the BRM 15.2 Installation to the Database Schema".
14. Upgrade the BRM database schema to the BRM 15.2 schema. See "Upgrading the BRM Database Schema to the BRM 15.2 Schema".
15. Upgrade the Pipeline Manager database schema to the BRM 15.2 schema. See "Upgrading the Pipeline Manager Database Schema to the BRM 15.2 Schema".
16. Install BRM 15.2 client applications. See "Installing the BRM 15.2 Client Applications".
17. Add optional components. See "Adding Optional Components".
18. Check other database configurations. See "Updating Configurations".
19. Upgrade Account Migration Manager for multischema systems. See "Upgrading AMM in a Multischema System".
20. Restore customizations in BRM 15.2. See "Restoring Customizations".

Preparing for the Upgrade

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 12.0:

1. Ensure that no users are logged in.

   Users include customers, client applications, customer service representatives (CSRs), and so on.

2. Stop all BRM 12.0 processes. For more information, see "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

   Only the database instances should be running during the installation.

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 12.0 Data

Creating a complete backup of the data in your BRM 12.0 environment requires the following actions to be performed in your BRM 12.0 environment:

1. Backing Up BRM Files
2. Backing Up Your BRM and Pipeline Manager 12.0 Databases

Backing Up BRM Files

Back up your BRM 12.0 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 12.0 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 12.0

> **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 your upgrade.

For more information, see "Transferring Customizations to the New Release".

Backing Up Your BRM and Pipeline Manager 12.0 Databases

> **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 when you import the backup data into the appropriate system for your new version.

Make a complete offline backup of your BRM database and, if you are using Pipeline Manager, also the Pipeline Manager database. Make the backup 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** or **expdb** utility to export all BRM 12.0 tables. This helps to restore individual tables, if necessary. For more information about the **exp** and **expdb** utilities, 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 the Database for Your New Version

> **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 15.2

This section describes the steps necessary to upgrade to BRM 15.2.

To upgrade to 15.2, perform the following tasks:

1. Importing Your BRM 12.0 Data
2. Setting the Environment Variables for Upgrade
3. Installing BRM and Pipeline Manager for Upgrade

Importing Your BRM 12.0 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 12.0), and then import the backup data you created in "Backing Up Your BRM and Pipeline Manager 12.0 Databases". 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 15.2.

Use the **imp** or **impdb** utility to import the backup data. For more information about the **imp** and **impdb** utilities, 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.

Setting the Environment Variables for Upgrade

Before installing the upgrade, ensure that the environment variables are set correctly. For information about the latest compatible versions of Java, Perl, and Kafka, see "BRM Software Compatibility" in *BRM Compatibility Matrix*.

To set the environment variables, perform the following for all users you are going to use to install or run the BRM server:

1. Set the JAVA_HOME environment variable to the directory in which the latest version of Java 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_381**.

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 Java version is not installed in the directory specified by JAVA_HOME.

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_38_0/linux.

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

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

5. If your BRM client wallet is not stored in the default location, set the BRM_CONF_WALLET and BRM_WALLET environment variables to point to the directory in which the BRM client wallet is stored:

   ```
   setenv BRM_CONF_WALLET brm_wallet_path
   setenv BRM_WALLET brm_wallet_path
   ```

6. Set the TNS_SERVICE_NAME environment variable to the service name for your primary database:

   ```
   setenv TNS_SERVICE_NAME service_name
   ```

7. 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 your current instance of BRM is installed.

8. If you will be using the Kafka Data Manager (DM), set the Kafka environment variables:

   > **Note**
   >
   > The Kafka DM is supported in BRM 12.0 Patch Set 4 and later releases.

   ```
   setenv KAFKA_HOME Kafka_path
   setenv KAFKA_BOOTSTRAP_SERVER_LIST KafkaHost1:port1,KafkaHost2:port2
   ```

   where:

- *Kafka_path* is the path to the directory in which the Kafka library JARs are installed.
- *KafkaHost1:port1,KafkaHost2:port2* are the hosts and ports that the Kafka client will connect to in a bootstrap Kafka cluster the first time it starts. You can specify any number of hosts and ports in this list.
  You can alternatively set this list in the **dm_kafka_config.xml** file. See "Mapping Business Events to Kafka Topics" in *BRM Developer's Guide*.

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 start the installer. See "Starting the BRM GUI Installer".

   The Welcome window appears.
3. Click **Next**.

   The Installation Location window appears.
4. Enter the full path or browse to the directory in which BRM 15.2 is going to be installed. Click **Next**.

   > ⚠ **Caution**
   >
   > Install BRM 15.2 in a different location from BRM 12.0 to avoid many upgrade problems.

   The Installation Type window appears.
5. Select **Custom**, and then click **Next**.

   The Feature Sets Selection window appears.
6. From the components list, select **Upgrade Manager Framework** and then select the components that you already installed in BRM 12.0. Click **Next**.

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

> **Note**
>
> - Ensure that you select the components in **Upgrade Manager Framework** (**Portalbase Upgrade** for BRM and **Pipeline Upgrade** for Pipeline Manager) that correspond to the components already installed in your BRM 12.0 or BRM 12.0 patch set environment. In the rest of the list, you must also select the same components that are already installed in your BRM 12.0 or BRM 12.0 patch set environment. You can install any additional optional component after completing the BRM 15.2 upgrade. See "Adding Optional Components" for instructions on installing the optional components.
> - If the existing BRM components are installed on separate machines, install the 15.2 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 15.2 server on the machine on which the existing BRM server is installed and BRM 15.2 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 window appears.

7. Specify the information in Table 16-1, and then click **Next**.

Table 16-1 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 Wallet Details window appears.

8. Enter the information in Table 16-2, and then click **Next**.

Table 16-2 Oracle Wallet Details

Field	Description
Wallet Password	Enter the password for the BRM 12.0 client wallet.
Wallet Location	For a standalone instance or a primary instance, enter the full path or browse to the directory in which the BRM client wallet for your 12.0 installation is located, for example, *BRM12_home*/**wallet/client**. For a secondary instance, enter the full path or browse to the directory in which the BRM client wallet or your primary instance is located, for example, *BRM15_home*/**wallet/client**.

9. In the Specify Wallet Details window, enter the details listed in Table 16-3 and then click **Next**.

Table 16-3 Specify Wallet Details

Field	Description
Client Wallet Password	The password for the client Oracle wallet. Use the same password used for 12.0. In a multischema system, this should be the same wallet password used in the primary schema for 12.0.
Confirm Client Password	This field is inactive for upgrade, since the password should be the same as the one for 12.0.
Root Wallet Password	The password for the root Oracle wallet
Confirm Root Password	The root Oracle wallet password again
Server Wallet Password	The password for the server Oracle wallet
Confirm Server Password	The server Oracle wallet password again

10. In the following windows, provide the requested information, and then click **Next**, with the following considerations for upgrade:

    - In the BRM Root User Details window, in the **Root Password** field, ensure that you enter the existing BRM root password that you provided during the BRM 12.0 installation.
    - In the Select Running the pin_setup Script window, select **Yes**. This will create the appropriate pin.conf files for your new environment. Because you selected upgrade packages to install, this will not initialize the database.
    - Although it is recommended that you stop all services before running the upgrade, you can have the BRM 12.0 environment running during the installation of BRM 15.2. If this is the case, or if the two environments are ever going to be running at the same time, ensure that you provide different port numbers for the feature sets than the ones used for BRM 12.0 to avoid port conflict.

    See "Installing All BRM Components" for descriptions of the fields displayed. Continue moving through the windows until the Installation Summary window appears.

    Your responses are written to the *BRM_home***/setup/pin_setup.values** file, where *BRM_home* is the directory in which the new BRM server software is installed.

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

    The Installation Progress window 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 window 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.

13. Click **Finish** to exit the installer.
14. Proceed to "Upgrading Your Data to BRM 15.2".

Upgrading Your Data to BRM 15.2

This section describes how to upgrade your data to BRM 15.2.

To upgrade your data to 15.2, perform the following tasks:

1. Pointing the BRM 15.2 Installation to the Database Schema
2. Upgrading the BRM Database Schema to the BRM 15.2 Schema
3. (If you are using Pipeline Manager) Upgrading the Pipeline Manager Database Schema to the BRM 15.2 Schema

Pointing the BRM 15.2 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 database upgrade script, verify that the Oracle Data Manager (DM) is configured correctly as this DM provides the interface to the BRM 15.2 database. The Oracle DM should point to the BRM 15.2 database in which you imported your BRM 12.0 backup data, as described in "Importing Your BRM 12.0 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 15.2 Schema

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

To upgrade the BRM 12.0 database schema to a BRM 15.2 database schema, complete one of the following:

- Upgrading the Schema on Single-Schema Systems
- Upgrading the Schema on Multischema Systems

Upgrading the Schema on Single-Schema Systems

To upgrade the BRM schema on single-schema systems:

1. Grant additional permission to the BRM database user:

   a. Connect to the primary database instance using SQL*Plus:

      ```
      % sqlplus system@databaseAlias as sysdba
      ```

   b. Run the following command:

      ```
      SQL> GRANT SELECT ON V_$SESSION TO BRM_user;
      Grant succeeded.
      SQL> commit;
      Commit complete.
      ```

   c. Exit SQL*Plus.

2. Edit the *BRM_home*/**setup/scripts/pin_setup.values** file and ensure that the following values are set:

   ```
   $SETUP_INIT_DB=YES
   $SETUP_DROP_ALL_TABLES=NO
   $CREATE_DATABASE_TABLES=NO
   ```

   Also set/verify the following values:

   - (Optional) Set 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.
   - (Optional) Set PIN_LOG_DIR to the directory in which you want to create the BRM log files.
   - Validate the other information in the file, especially the information about the database.

3. Save and close the file.

4. Edit the *BRM_home*/**setup/pin_tables.values** file and set the following values:

   ```
   $PIN_CONF_PARTITION_IND             = "local";
   $PIN_CONF_NON_EVENT_PARTITION_IND   = "local";
   ```

5. Save and close the file.

6. Run the **source** command on the **source.me** file:

   - For Bash Shell:

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

   - For C shell:

     ```
     source source.me.csh
     ```

7. 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.
> - Ensure that the WALLET_FILE_STORAGE_T table is present in the database before running this script.

```
perl script
```

where *script* is **pin_upgrade_15.pl** for BRM 15.2.0.

> ⓘ **Note**
>
> Do not press any button until the command prompt asks for the BRM wallet password in the next step. Otherwise, the script will consider whatever key you have pressed to be the BRM wallet password.

8. At the "Enter Password for the wallet" prompt, enter the BRM client wallet password (which should be the same for both 12.0 and 15.2) and wait for the script to finish.
9. 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, perform the tasks in "Upgrading the Schema on Single-Schema Systems" on your primary schema.

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

1. Grant an additional permission to the BRM database user:
   a. Connect to the primary database instance using SQL*Plus:

      ```
      % sqlplus system@databaseAlias as sysdba
      ```

   b. Run the following command:

      ```
      SQL> GRANT SELECT ON V_$SESSION TO BRM_user;
      Grant succeeded.
      SQL> commit;
      Commit complete.
      ```

   c. Exit SQL*Plus.
2. Before you upgrade the secondary schema, set the BRM_WALLET environment variable to the BRM wallet in the primary schema:

   ```
   setenv BRM_WALLET BRM_home/wallet/client
   ```

3. Perform the tasks in "Upgrading the Schema on Single-Schema Systems" on your secondary schema.

To configure your entire multischema system, perform the following on the appropriate BRM instances:

1. On the primary BRM instance, edit the *BRM_home***/apps/multi_db/config_dist.conf** file and add a line specifying the schema name at the end of each block in the file. For example, if you have two database schemas, your file might look like this:

   ```
   DB_NO = "0.0.0.1" ;        # 1st database config block.
   PRIORITY = 1 ;
   MAX_ACCOUNT_SIZE = 100000 ;
   STATUS = "OPEN" ;

   DB_NO = "0.0.0.2" ;        # 2nd database config block.
   PRIORITY = 1 ;
   MAX_ACCOUNT_SIZE = 100000 ;
   STATUS = "OPEN" ;
   ```

   Update the file to look more like the following:

   ```
   DB_NO = "0.0.0.1" ;        # 1st database config block.
   PRIORITY = 1 ;
   MAX_ACCOUNT_SIZE = 100000 ;
   STATUS = "OPEN" ;
   SCHEMA_NAME = "pin01" ;

   DB_NO = "0.0.0.2" ;        # 2nd database config block.
   PRIORITY = 1 ;
   MAX_ACCOUNT_SIZE = 100000 ;
   STATUS = "OPEN" ;
   SCHEMA_NAME = "pin02" ;
   ```

   using the correct schema names for your environment.

2. Edit the *BRM_home***/setup/scripts/pin_multidb.conf** file in the primary BRM instance and set the following values for each secondary database:

   ```
   #-------------------------
   # Secondary db information
   #-------------------------

   $PIN_MD_SECONDARY_DBNO     [0] = "db_no";
   $PIN_MD_SECONDARY_OWNER    [0] = "sec_owner";
   #$PIN_MD_SECONDARY_PASSWD  [0] = "pinb";
   $PIN_MD_SECONDARY_DBNAME   [0] = "sec_db";
   $PIN_MD_SECONDARY_SQLHOST  [0] = "";
   $PIN_MD_SECONDARY_HOSTNAME [0] = "sec_hostname";
   $PIN_MD_SECONDARY_PORT     [0] = "dm_port";       # dm port
   ```

   where:

   - *db_no* is the database number of the secondary schema, for example **0.0.0.2**.
   - *sec_owner* is the owner of the secondary database.
   - *sec_db* secondary database name.
   - *dm_port* port for the DM.

3. Edit the DM **pin.conf** file on the primary BRM instance and set the following value for each secondary schema:

   ```
   - dm schema db_no sec_owner
   ```

   where:

   - *db_no* is the database number of the secondary schema, for example **0.0.0.2**.
   - *sec_owner* is the owner of the secondary database.

4. Edit the DM **pin.conf** file on each secondary BRM instance and set the following values for the primary schema:

   ```
   - dm schema 0.0.0.1 prim_owner
   ```

   where:

   - *prim_owner* is the owner of the primary database.

5. Stop and restart all of the services on the primary BRM instance and all secondary instances.

6. On the primary BRM instance, connect to SQL*Plus and run the following scripts:

   ```
   @create_procedures_character_set.plb
   grant_permissions_oracle.plb
   grant_permissions('sec_owner')
   ```

   where:

   - *character_set* specifies the database character set, either **UTF8** or **AL32UTF8**.
   - *sec_owner* is the owner of the secondary schema.

   Run the last command for the owner of each secondary schema.

7. On each secondary BRM instance, connect to SQL*Plus and run the following scripts:

   ```
   @create_procedures_character_set.plb
   grant_permissions_oracle.plb
   grant_permissions('prim_owner')
   ```

   where:

   - *character_set* specifies the database character set, either **UTF8** or **AL32UTF8**.
   - *prim_owner* is the owner of the primary schema.

8. On the primary BRM instance, do the following for *each* secondary schema:

   a. Edit the *BRM_home***/sys/dd/data/device_framework_primary_schema_75ps4.sql** script and update all instances of **SEC_SCHEMA_NAME** to the name of the secondary database schema owner.

   b. Connect to the primary database schema and run the edited script in SQL*Plus.

9. On *each* secondary BRM instance, connect to the secondary database schema and run the *BRM_home***/sys/dd/data/device_framework_secondary_schema_75ps4.sql** script in SQL*Plus.

Upgrading the Pipeline Manager Database Schema to the BRM 15.2 Schema

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

To upgrade the Pipeline Manager database schema:

1. Create a temporary directory for use during the upgrade by going to *Pipeline_home***/upgrade** and creating the *Pipeline_home***/upgrade/tmp** directory.
2. Update the *Pipeline_home***/upgrade/pipeline_upgrade.cfg** file and set the following values:

   ```
   $PIN_TEMP_DIR = "temp_path"
   $MAIN_DB{'user'} = "db_user";
   $PIPELINE_TBLSPACE = "tablespace pipeline_tablespace";
   ```

   where:

   - *temp_path* is the directory you just created.
   - *db_user* is the primary database user.
   - *pipeline_tablespace* is the tablespace containing pipeline data.

   Ensure that you also update the database information as appropriate.
3. Grant the required access to the user **pin** on the Pipeline Manager tables and sequences, if you have not already done so. See "Loading the Tailor-Made Stored Procedure" for more information about the Pipeline Manager tables and sequences that you should grant user **pin** access to.
4. 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.
   > - Ensure that the WALLET_FILE_STORAGE_T table is present in the database before running this script.

   ```
   perl script
   ```

   where *script* is **pin_upgrade_15.pl** for BRM 15.2.0.

   This script runs a series of scripts that upgrade Pipeline Manager data from 12.0 to 15.2. The log files will be located in the temporary directory you created.
5. In SQL*Plus, run the *Pipeline_home***/database/Oracle/Scripts/create_pricing_discountmodel_procedures.plb** script in the Pipeline schema.

Completing the Upgrade

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

- Creating Large Indexes
- Installing the BRM 15.2 Client Applications
- Adding Optional Components
- Updating Configurations
- Upgrading AMM in a Multischema System
- Restoring Customizations

Creating Large Indexes

For performance reasons, you must create some indexes manually after the upgrade.

To create the indexes in the default tablespace, use the following commands:

```
CREATE INDEX i_event_item_tr_arbill__id ON event_item_transfer_t
(ar_bill_obj_id0);
CREATE INDEX i_event_session_obj__id ON event_t (session_obj_id0);
CREATE INDEX i_purchased_discount_acc_obj__id ON purchased_discount_t
(account_obj_id0);
CREATE INDEX i_purchased_product_acc_obj__id ON purchased_product_t
(account_obj_id0);
CREATE UNIQUE INDEX i_billinfo_id_code__id ON billinfo_t
(identification_code);
```

To create the indexes in another tablespace (for example, the index tablespace), use the following commands:

```
CREATE INDEX i_event_item_tr_arbill__id ON event_item_transfer_t
(ar_bill_obj_id0) tablespace tablespace_name;
CREATE INDEX i_event_session_obj__id ON event_t (session_obj_id0) tablespace
tablespace_name;
CREATE INDEX i_purchased_discount_acc_obj__id ON purchased_discount_t
(account_obj_id0) tablespace tablespace_name;
CREATE INDEX i_purchased_product_acc_obj__id ON purchased_product_t
(account_obj_id0) tablespace tablespace_name;
CREATE UNIQUE INDEX i_billinfo_id_code__id ON billinfo_t
(identification_code) tablespace tablespace_name;
```

where *tablespace_name* is the name of the tablespace that should contain the indexes.

Installing the BRM 15.2 Client Applications

When you install client applications, be sure to update the BRM 15.2 **Infranet.properties** and **INI** files with any 15.2 customizations. See "Installing BRM Thick Clients" for more instructions.

> **Note**
>
> You must upgrade *all* existing client applications and optional components to BRM 15.2 before you install additional client applications and optional components in BRM 15.2.

To upgrade *custom* client applications, recompile them with BRM 15.2 libraries. See "Updating Custom Applications" for more information.

Adding Optional Components

At this point, add the optional components that you plan to use in the new version of BRM. See "Installing Individual BRM Components" for instructions.

> **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 new BRM environment.

Update the Default Settings for the pin_rel Utility

If you use Rated Event (RE) Loader, ensure that the database configuration 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 12.0 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. If your system includes Pipeline Manager, follow the instructions here: "Loading the pin_notify File"
2. Copying Files for Compiling CMs
3. Migrating Custom Code to a 64-bit Environment
4. Updating Queue Handling
5. Setting the Environment Variables for Pipeline Manager
6. Configuring SSL for the BRM Database
7. Removing Invalid Objects from the Database
8. Modifying the Root Encryption Key
9. If you have upgraded from BRM 12.0 Patch Set 4 or earlier, follow the instructions here: "Setting the Bill State for Preexisting Bills"
10. If you have upgraded from BRM 12.0 Patch Set 4 or earlier, follow the instructions here: "Migrating Tax Codes"
11. If you have upgraded from BRM 12.0 Patch Set 4 or earlier, follow the instructions here: "Updating pin_inv_doc_gen Infranet.properties"
12. Removing the TRIG_CYCLE_DEFERRED_TAX Trigger

13. If you intend to migrate legacy account information into BRM, follow the instructions here: "Preparing to Migrate Legacy Account Data"

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:

1. Go to the *BRM_home*/**sys/data/config** directory.
2. 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.
3. Stop and restart the Connection Manager (CM).

Copying Files for Compiling CMs

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

To copy the required files for compiling CMs:

1. Copy the following files from the *BRM_home*/**lib** directory to the *BRM_home*/**PortalDevKit/source/sys/cm** directory:
   - **libcmpin.so**
   - **libdmpin.so**
2. Copy the **cm.cpp** file from the *BRM_home*/**source/sys/cm** directory to the *BRM_home*/**PortalDevKit/source/sys/cm** directory.
3. Open the *BRM_home*/**PortalDevKit/source/samples/env.unix** file in a text editor.
4. Add or modify the following environment variables:

   ```
   LIBDIR = BRM_home/PortalDevKit/lib
   RW_INCDIR = PortalDevKit_hostname/rwWorkspace
   INCDIR = BRM_home/PortalDevKit/include
   PCM_JAR = BRM_home/jars/pcm.jar
   JDK_HOME = jdk_path
   PCMEXT_JAR = BRM_home/jars/pcmext.jar
   ```

   where:
   - *hostname* is the name of the machine on which the BRM server is installed.
   - *jdk_path* is the path to the directory in which the latest version of JRE certified with BRM is installed, such as /Linux/x86_64/packages/jdk/jdk1.8.0_381.

5. Save and close the file.

Migrating Custom Code to a 64-bit Environment

To migrate the code for your policy opcodes to 64-bit.

1. If you are using custom makefiles, make the following changes in each:
   - If the makefile contains **-m32**, remove it.
   - Remove **elf_i386** from the file, or replace it with **elf_x86_64**.
   - Update the dynamic linking path for 64-bit. For example, change **-shared -L/usr/lib** to **-shared -L/usr/lib64**
2. Change the format specifiers for variables whose sizes are different for 64-bit. For specifics, see Table 16-4.

Table 16-4 Changes in Data Type Format Between 32-bit and 64-bit

Data type	32-bit Length	64-bit Length
time_t	4 bytes	8 bytes
Pointer	4 bytes	8 bytes
long	4 bytes	8 bytes
wchar_t	2 bytes	4 bytes
size_t	4 bytes	8 bytes

> **Note**
>
> The int and int32 data types remain 4 bytes, and the int64 data type remains 8 bytes.

3. Make specific changes relating to time/date formats, to allow for dates beyond the year 2038.
   - If your code ever uses the **int** data type for a date, change the type to **time_t**. While the **int** data type remains at 4 bytes, the **time_t** data type has increased to 8 bytes.
   - When your code calls any PCM library function, such as PIN_FLIST_FLD_SET or PIN_FLIST_FLD_GET, use variables of type **time_t** or **pin_fld_tstamp_t** for any PIN_FLDT_TSTAMP fields.
4. Enable the **-Wconversion** flag during compilation so that it will report any warnings related to implicit type conversions. Solve all of those errors. In particular, make sure you solve any Wpointer-to-int-cast warnings. These might be generated by pointers that assume the old data type sizes.
5. Be aware of the magic numbers and any Bit Shifting Operations being used in the code. These may need to be changed. For example, some issues with magic numbers are:

```
longvar = 0xffffffff; //-1 in 32-bit environment
longvar = 0xffffffff; //4294967295 in 64-bit environment
longvar = 0xffffffffffffffff; //-1 in 64-bit environment
```

Updating Queue Handling

The **dm_ifw_sync** and **dm_aq** queues were previously used for Advanced Queuing in BRM. However, the Account Sync Manager (dm_ifw_sync) and the Queue Manager (dm_aq) components were deprecated in BRM 15.0. Their functionality has been moved to **dm_oracle**, which has been enhanced to enqueue these messages going forward.

The *BRM_home***/apps/pin_ifw_sync/** and *BRM_home***/apps/pin_aq/** directories are not present in BRM 15.0 or later. They have been replaced by a new directory, *BRM_home***/apps/pin_publish_aq/**. This directory contains the **pin_publish_oracle.pl** utility, which is used to maintain Advanced Queues meant to publish messages for enterprise integration and the Pipeline Rating Engine.

To update your system for this change:

1. Update the payload configuration XML files located in *BRM_home***/sys/eai_js** to change the publisher database number. Change the numbers **0.0.9.7** (dm_aq) and **0.0.9.9** (dm_ifw_sync) to **0.0.0.0**. This will route the events to **dm_oracle**.
2. Update the business event queue mapping by updating the *BRM_home***/sys/dm_oracle/aq_event_map** file with the information that was previously directed to **dm_ifw_symc** and **dm_aq**. This file is equivalent to the previous **ifw_sync_queuenames** file.

   The DM pin.conf file has been updated to contain the following parameter:

   ```
   - dm aq_event_map_file ./aq_event_map
   ```

   You can use this parameter to change the name of the queue file according to your requirements.

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 12.0, you must set some environment variables for Pipeline Manager.

On the machine on which you installed the new version of the BRM 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

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.

Removing Invalid Objects from the Database

After installing the new version of BRM, 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**.

Modifying the Root Encryption Key

To enhance security, you should modify the root encryption key on a regular basis.

See the appropriate section for more information:

- Modifying the Root Encryption Key on Single-Schema Systems
- Modifying the Root Encryption Key on Multischema Systems

Modifying the Root Encryption Key on Single-Schema Systems

To modify the root encryption key on single-schema systems:

1. Back up the CRYPTKEY_T and WALLET_FILE_STORAGE_T tables in the BRM database. See your database documentation for more information on performing database table and view backups.

   > **Note**
   >
   > Store the backup in a safe location. The data in the backed up CRYPTKEY_T table and the root key wallet will become necessary if you encounter any issues in modifying the root encryption key.

2. Go to the directory in which you installed the BRM server, and edit the appropriate source.me file:

   a. For Bash shell, edit the **source.me.sh** file and look for the following text:

      ```
      export BRM_WALLET
      ```

      Update the wallet location to read:

      ```
      export BRM_WALLET=$PIN_HOME/wallet/
      ```

   b. For C shell, edit the **source.me.csh** file and look for the following text:

      ```
      setenv BRM_WALLET
      ```

      Update the wallet location to read:

      ```
      setenv BRM_WALLET $PIN_HOME/wallet/
      ```

3. Source the file you just edited:

   a. Bash shell:

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

   b. C shell:

      ```
      source source.me.csh
      ```

4. Run the following command from the *BRM_home*/**sys/test** directory.

   ```
   pin_crypt_app -genrootkey
   ```

5. Ensure that the **testnap** utility works.

6. Run the following command from the *BRM_home*/**sys/test** directory.

   ```
   pin_crypt_app -useZT -genkey -update_dm_conf
   ```

7. Restart the CM and DM and ensure that the **testnap** utility works.

Modifying the Root Encryption Key on Multischema Systems

To modify the root encryption key on multischema systems:

1. Back up the CRYPTKEY_T and WALLET_FILE_STORAGE_T tables in the BRM database in the primary schema and the CRYPTKEY_T table and WALLET_FILE_STORAGE_T view in all of the secondary schemas. See your database documentation for more information on performing database table and view backups.

   > **Note**
   >
   > Store the backup in a safe location. The data in the backed up CRYPTKEY_T table and the root key wallet will become necessary if you encounter any issues in modifying the root encryption key.

2. Go to the directory in which you installed the BRM server on the primary and all secondary instances, and edit the appropriate source.me file:

   a. For Bash shell, edit the **source.me.sh** file and look for the following text:

      ```
      export BRM_WALLET
      ```

      Update the wallet location to read:

```
export BRM_WALLET=$PIN_HOME/wallet/
```

   b. For C shell, edit the **source.me.csh** file and look for the following text:

```
setenv BRM_WALLET
```

   Update the wallet location to read:

```
setenv BRM_WALLET $PIN_HOME/wallet/
```

3. Source the file you just edited:

   a. Bash shell:

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

   b. C shell:

```
source source.me.csh
```

4. Verify that the key value in CRYPTKEY_T is the same for both the primary and secondary schemas. If it is not:

   a. Copy the value of **pin_crypt_aes4dm** from *BRM_home***/sys/dm_oracle/pin.conf** to the same location on all of the secondary instances.

   b. Restart the BRM services on the primary and secondary BRM instances.

   c. Run the following command from the *BRM_home***/sys/test** directory in the primary instance:

```
pin_crypt_app -useZT -genkey -update_dm_conf
```

   d. Copy the value of **pin_crypt_aes4dm** from *BRM_home***/sys/dm_oracle/pin.conf** to the same location on all of the secondary instances.

   e. Restart the BRM services on the primary and secondary BRM instances.

   f. Run the following command from the *BRM_home***/apps/pin_crypt** directory in the primary instance:

```
pin_crypt_upgrade -a
```

   g. Ensure that the **testnap** utility works.

5. If the key value is the same for both the primary and secondary schemas, update the root key by doing the following:

   a. Run the following command from the *BRM_home***/sys/test** directory:

```
pin_crypt_app  -genrootkey
```

   b. Ensure that the **testnap** utility works.

   c. Run the following command from the *BRM_home***/sys/test** directory:

```
pin_crypt_app -useZT -genkey -update_dm_conf
```

   d. Restart the CM and DM in the primary instance and ensure that the **testnap** utility works.

   e. Copy the value of **pin_crypt_aes4dm** from *BRM_home***/sys/dm_oracle/pin.conf** to the same location on all of the secondary instances.

   f. Verify that the key value in CRYPTKEY_T is the same for both the primary and secondary schemas.

   g. Restart the CM and DM in the secondary instances.

   h. Ensure that the **testnap** utility works.

Setting the Bill State for Preexisting Bills

If you have upgraded from BRM 12.0 Patch Set 4 or earlier, after the upgrade any new **/bill** objects include the PIN_FLD_STATE field set to the appropriate value. For all **/bill** objects that existed in the system before applying the BRM patch set, this field will be set to UNDEFINED.

You can optionally run a stored procedure to update the value of PIN_FLD_STATE on preexisting bills. This has no impact on functionality but can be done for consistency.

To run the stored procedure:

1. Go to the *BRM_home***/sys/dd/data** directory.
2. Connect to the Oracle database with 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 this command:

   ```
   SQL> @update_bill_state_12PS5.source
   ```

The value is set according to the following conditions:

Condition	PIN_FLD_STATE Value
The value of PIN_FLD_END_T is 0	INPROGRESS
The value of PIN_FLD_DUE is not zero and is greater than or equal to the value of PIN_FLD_CURRENT_TOTAL	NEW
The value of PIN_FLD_DUE is not zero and is less than the value of PIN_FLD_CURRENT_TOTAL	PARTIALLYPAID
The value of PIN_FLD_DUE is zero.	SETTLED

See "About Bill States" in *BRM Concepts* for more information about these bill states.

Migrating Tax Codes

If you have upgraded from BRM 12.0 Patch Set 4 or earlier, after the upgrade tax codes are stored in the **/config/taxcodes_map** object rather than in the **taxcodes_map** file.

If you are upgrading from BRM 12.0 Patch Set 4 or earlier, the Patch Set installer loads the **/config/taxcodes_map** object with default values. You must migrate any existing tax codes from the **taxcodes_map** file to the **/config/taxcodes_map** object.

To migrate your existing tax codes to so, perform the following steps:

1. If your existing BRM system contains multiple **taxcodes_map** files, collate them into a single file.

2. Run the **convert_taxcode.pl** script to convert your existing tax codes into XML format:

   ```
   perl BRM_home/setup/scripts/convert_taxcode.pl inputFile [outputFile]
   ```

   where:

   - *inputFile* is the name and location of the file that contains your tax code configurations from Patch Set 4 or earlier. By default, these tax codes are stored in the *BRM_home***/sys/cm/taxcodes_map** file.
   - *outputFile* is the name of the output file to generate. This file will contain your tax codes converted into XML format. If not provided, the output file will be named **config_taxcodes_map.xml**.

3. Copy the generated **config_taxcodes_map.xml** file to the *BRM_home***/sys/data/config** directory.

4. Load the **config_taxcodes_map.xml** file into the BRM database:

   ```
   BRM_home/apps/load_config/load_config -M BRM_home/sys/data/config/
   config_taxcodes_map.xml
   ```

   See "load_config" in *BRM Developer's Guide* for information about the utility's syntax and parameters.

Your tax codes have been loaded into the **/config/taxcodes_map** object.

Updating pin_inv_doc_gen Infranet.properties

If you are upgrading from BRM 12.0 Patch Set 4 or earlier, after the upgrade entries in the **pin_inv_doc_gen Infranet.properties** file have changed.

Copy the changes from the *BRM_home***/apps/pin_inv_doc_gen/Infranet.properties.sample** file to your *BRM_home***/apps/pin_inv_doc_gen/Infranet.properties** file, and then set the following new entries:

- **infranet.schedulerdb.url**: Specifies the scheduler database URL in the following format:

  ```
  jdbc:oracle:thin:hostname:port/service
  ```

  where *hostname* is the hostname of the scheduler database, *port* is the port number for the scheduler database, and *service* is the service name of the scheduler database.
- **infranet.schedulerdb.user**: Specifies the user name for the scheduler database.
- **infranet.schedulerdb.credentials**: Specifies the security credentials for connecting to the scheduler database.
- **infranet.jdbcpool.size**: Specifies the initial number of connections maintained in the pool. The default is set to the same as **burst.threadpool.size**.
- **infranet.jdbcpool.maxsize**: Specifies the maximum number of connections that can be created. The default is set to the same as **burst.threadpool.maxsize**.

Removing the TRIG_CYCLE_DEFERRED_TAX Trigger

Previous versions of BRM used the TRIG_CYCLE_DEFERRED_TAX trigger, but it was later disabled. If you see it in your environment, remove it to avoid miscalculation of cycle taxes.

To remove the trigger, use SQL*Plus to run the following command on the BRM database as a user with sysdba privileges:

```
drop trigger TRIG_CYCLE_DEFERRED_TAX;
```

Preparing to Migrate Legacy Account Data

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

If you intend to migrate legacy account information into BRM:

1. Go to the *BRM_home***/sys/dd/data** directory.
2. Open SQL*Plus as the BRM database user.
3. Run this command:

   ```
   SQL> @update_due_amount_oracle.sql
   ```

17

Upgrading BRM 15.x to BRM 15.2

Learn how to upgrade Oracle Communications Billing and Revenue Management (BRM) 15.0 or 15.1 to BRM 15.2 and to roll back the upgrade.

Topics in this document:

- About Upgrading to the Later Release
- Performing a Zero Downtime Upgrade
- Tasks Involved in the Upgrade Process for a New Release
- Preparing for the Upgrade to the New Release
- Upgrading to the New Release
- Post-Upgrade Procedures for the New Release
- Installing Optional Components After the Release Upgrade

> **Note**
>
> - To upgrade an existing BRM 12.0 or 12.0 patch set installation to BRM 15.2, see "Upgrading BRM 12.0 to BRM 15.2". For information on the supported upgrades, see *BRM Compatibility Matrix*.
> - To do a full installation of BRM 15.2, see "Installing BRM".

About Upgrading to the Later Release

> **Note**
>
> Test the later release on a non-production system before you deploy it on a production system.

The BRM installer allows you to upgrade from BRM 15.0 or 15.1 or a later BRM 15.0 or 15.1 release to BRM 15.2. For example, you could upgrade a BRM 15.0.1 system to BRM 15.2.

When you do an upgrade, only files that have been changed are updated. The installer makes a backup of any file it updates. You use the backup files to merge your customizations and to uninstall the later release. The pre-upgrade version of each updated file is stored with the extensions GA and TGA and is left in its original directory. For example, when you install BRM 15.2 to upgrade Pipeline Manager, the existing **sample.reg** file is renamed to **sample.reg.TGA**.

The installer contains multiple packages to upgrade different BRM and Pipeline Manager components. You can install or upgrade these components by installing their corresponding packages:

- BRM
- BRM SDK
- Pipeline Manager

Performing a Zero Downtime Upgrade

> **Note**
>
> - If you perform the zero downtime upgrade, billing might fail for some accounts. In that case, rerun billing for the failed bill units by running the billing utility with the **-retry** option.
> - If your upgrade fails due to resource busy or timeout errors in the database, consult your DBA about increasing the value of the DDL_LOCK_TIMEOUT parameter for your database.

You can use the zero downtime upgrade method to upgrade your existing BRM installation with very minimal disruption to your existing installation and the services that are provided to your customers.

Before you perform the zero downtime upgrade, ensure the following:

- You have two instances of the existing BRM 15.0 or 15.1 release on your system; for example, two instances of BRM 15.0.1.
- Both BRM instances are currently running. This includes the primary and secondary instances and all the applications in the components connected to your BRM system.
- Both BRM instances are connected to the same BRM and Pipeline Manager database schemas.

To perform the zero downtime upgrade:

> **Note**
>
> Do not shut down the applications in the components connected to your BRM system while the upgrade is running.

1. Route all the traffic (for example, phone calls or data usage) from the primary instance to the secondary instance.
2. Upgrade the *primary* instance and the BRM database schema to the new release. See "Upgrading to the New Release" for instructions.
3. Start the *primary* instance. See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide* for instructions.
4. Reroute all traffic from the secondary instance to the primary instance.

> ⚠ **Caution**
>
> During the database upgrade on the *primary* instance, ensure that the services or operations in the *secondary* instance are running until all of the requests are rerouted to the *primary* instance.

5. Upgrade the *secondary* instance to the new release. See "Upgrading to the New Release" for instructions.

> ⓘ **Note**
>
> Because you have already upgraded the BRM database schema during the *primary* instance upgrade, you can skip "Upgrading the BRM Database Schema to the New Release".

6. Perform post-upgrade tasks on both *primary* and *secondary* instances. See "Post-Upgrade Procedures for the New Release" for instructions.

> ⓘ **Note**
>
> Batch operations, such as billing, might fail for a few accounts during the upgrade. You must restart the applications to complete your batch operations.

Tasks Involved in the Upgrade Process for a New Release

This section provides a list of tasks required to upgrade from BRM 15.0 or 15.1 to BRM 15.2. Some tasks are optional or apply only to certain 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 before upgrading your BRM system. See "Preparing for the Upgrade to the New Release" for more information.

1. Shut down the current instance of BRM.
2. Turn off service authentication and authorization.
3. Create a complete backup of your BRM data.
4. Install the latest compatible version of Java if it is different from the one for your current release.
5. Install the latest compatible version of Perl if it is different from the one for your current release.
6. Install any optional components you want to add to the old release. It is recommended to add any optional components to the instance you are upgrading from before the upgrade, because you cannot add them while you are doing the upgrade.

Perform the following tasks to upgrade to the new release. See "Upgrading to the New Release" for more information.

1. Set the environment variables, for example setting the variables for Java and Perl variables to the latest compatible version.
2. Verify that the BRM wallet location in **sqlnet.ora** is correct.
3. Obtain the BRM software.
4. Install the upgrade package.
5. Upgrade Business Operations Center.
6. Upgrade the BRM database schema to the schema for the new release.
7. If you are using Pipeline Manager, upgrade the Pipeline Manager database schema for the new release.
8. Install BRM client applications.
9. Add customizations.

After you upgrade to BRM 15.2, perform the following post-upgrade tasks. See "Post-Upgrade Procedures for the New Release" for more information.

1. Configure SSL for the BRM Database.
2. Remove invalid objects from the database.

Preparing for the Upgrade to the New Release

Complete these tasks before upgrading your BRM system:

1. Shut down the current instance of BRM. See "Shutting Down the Current Instance" for instructions.
2. Create a complete backup of your BRM data. See "Backing Up Your BRM and Pipeline Manager Databases" for instructions.
3. Install the latest compatible version of Java if it is different from the one for your current release. See "Installing Java" for instructions.
4. Install the latest compatible version of Perl if it is different from the one for your current release. See "Installing Perl" for instructions.
5. Install any optional components you want to add to the old release. See "Installing Optional Components" for instructions.

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 the existing BRM instance:

1. Ensure that no users are logged in.

Users include customers, client applications, customer service representatives (CSRs), and so on.

2. Stop all BRM processes. For more information, see "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

   Only the database instances should be running during the upgrade.

Backing Up Files

> **Note**
>
> In multischema systems, perform this task first on the primary BRM installation machine and then on the secondary BRM installation machines.

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

- All customized files, including any associated source code
- Registry files
- Policy files
- **start_all**
- **pin.conf**
- **pin_ctl.conf**
- **pin_setup.values**
- **Infranet.properties**
- All other files that contain any customizations used in your system

If any libraries that were used in a previous release are no longer needed for a later release, these libraries will be moved automatically to *BRM_home***/deleted_files** for reference, so you do not need to back them up manually.

Backing Up Your BRM and Pipeline Manager Databases

> **Note**
>
> In multischema systems, perform this task first on the primary database schema and then on the secondary database schemas.

Make a complete offline backup of your BRM database and, if you are using Pipeline Manager, also the Pipeline Manager database. Make the backup 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.

ⓘ **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.

Installing Optional Components

ⓘ **Note**

In multischema systems, perform this task first on the primary BRM installation machine and then on the secondary BRM installation machines.

Before upgrading, install any optional components that you want to add to your BRM system. For example, if you want to add Collections Manager to your BRM system, install Collections Manager in your old release before you install the BRM 15.2 release.

See "Installing Individual BRM Components" for more information.

Upgrading to the New Release

Perform the following tasks to upgrade to the new release:

⚠ **Caution**

When upgrading a multischema system, pay close attention to the system on which each task is performed.

1. Set the environment variables, for example setting the variables for Java and Perl variables to the latest compatible version. See "Setting the Environment Variables for Upgrade" for instructions.
2. Verify that the BRM wallet location in **sqlnet.ora** is correct. See "Setting BRM Wallet Location in sqlnet.ora" for instructions for updating this value.
3. Obtain the BRM software. See "Downloading the BRM Software" for instructions.
4. Install the upgrade package. See "Upgrading BRM for the New Release" or "Upgrading BRM for the New Release in Silent Mode" for instructions.
5. Upgrade the BRM database schema to the BRM 15.2 schema. See "Upgrading the BRM Database Schema to the New Release" for instructions.
6. Upgrade the Pipeline Manager database schema to the BRM 15.2 schema. See "Upgrading the Pipeline Manager Database Schema to the New Release" for instructions.
7. Upgrade Business Operations Center. See "Upgrading Business Operations Center" in *Business Operations Center Installation Guide* for instructions.
8. Install BRM 15.2 client applications. See "Installing the BRM 15.2 Client Applications" for instructions.
9. Add customizations. See "Adding Customizations" for instructions.

Upgrading BRM for the New Release

> **Note**
>
> In multischema systems, perform this task first on the primary BRM installation machine and then on the secondary BRM installation machines.

To upgrade to the new release:

1. Go to the directory where you downloaded the BRM installer, and start the installer. See "Starting the BRM GUI Installer".
2. In the Welcome window, click **Next**.
3. In the Installation Location window, enter the full path or browse to the directory in which you installed BRM 15.0 and then click **Next**.
4. In the Specify Prerequisite Libraries Location window, specify the following library information and then click **Next**.

Table 17-1 Fields in the Specify Prerequisite Libraries Location Window

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	You must set this to the same value used for the instance you are upgrading from.

5. In the Oracle Wallet Details window, enter the following information for accessing the Oracle wallets in BRM and then click **Next**.

Table 17-2 The Fields in the Oracle Wallet Details Window

Field	Description
Wallet Password	Enter the password for the client Oracle wallet for the existing 15.0 installation.
Wallet Location	Enter the path or browse to the directory in which the client Oracle wallet for the existing 15.0 installation is located.

6. In the Available Product Components window, select the components to install, and deselect any other components that you do not want to install. Click **Next.**

> **Note**
>
> You cannot deselect a component if it is required to install any of the selected components.

If you are installing an optional component for the first time, and you see an error similar to the following, see "Problem: An Error Occurred When Selecting an Optional Component to Install".

```
The distribution Billing Revenue Management 15.2.0.0.0 contains incompatible
features with the following:
Billing Revenue Management 15.0.1.0.0 [CORE 15.2.0.0.0->CORE 15.0.1.0.0 (CORE
15.2.0.0.0->[CORE 15.0.1.0.0])]
```

7. In the Database SSL Options window, select the same value (**Yes One Way**, **Yes Two Way**, or **No**) that was used for the instance you are upgrading from.
8. In the following windows, provide the requested information, and then click **Next**.
9. In the Installation Summary window, review the selections you have made in the preceding windows and then click **Install**.

   The Installation Progress window appears.

   > **Note**
   >
   > After installation begins, clicking **Cancel** stops the installation process, but the files that are already copied are not removed. You must manually remove the files.

10. When the installation is done, click **Next**. The Installation Complete window appears.

    Note the provided URL. You will use this URL to access BRM.
11. Click **Finish** to complete and exit.

Upgrading BRM for the New Release in Silent Mode

If you want to upgrade the BRM release, using the same configuration, in multiple environments. Silent install mode does not use the GUI and it runs in the background.

In this mode, you use a response file template that contains a predefined set of values to upgrade BRM to the new release. You can generate a response file that contains the parameters and values during the GUI installation.

Creating a Response File

To create a response file:

1. Create the response file by doing one of these:
   - Create a copy of the response file that was generated during the GUI upgrade. See "Upgrading BRM for the New Release" for more information.

     > **Note**
     >
     > The GUI Installer does not store passwords provided during installation in the response file. You must manually add the passwords after creating a copy of the response file.

   - Create a response file using the template by running the following command:

     ```
     Java_home/bin/java -jar jarFile -getResponseFileTemplates
     ```

     where:
     - *Java_home* is the directory in which you installed the latest compatible Java version.

  - *jarFile* is the BRM installer file. For example:

    ```
    brmserver_15.1.0.0.0_linux_generic_full.jar
    ```

  A response file is created with the default values.

  You can create as many response files as needed.

2. Open the file in a text editor.
3. Modify the response file you copied by specifying the key-value information for the parameters you want in your installation.

   > **Note**
   >
   > - The response file template contains guidelines and examples on how to enter the values in the parameters.
   > - The Installer treats incorrect context, format, and type values in a response file as if no value were specified.

4. Save and close the response file.

Upgrading the Release in Silent Mode

To upgrade the BRM release in silent mode:

1. Create the response file. See "Creating a Response File".
2. Copy the response file you created to the machine on which you run the silent installation.
3. On the machine on which you are running the silent installation, go to the directory where you downloaded the BRM software, and run the following command:

   ```
   Java_home/bin/java -jar jarFile -debug -invPtrLoc Inventory_home/oraInventory/
   oraInst.loc [parameter=value] -responseFile path -silent
   ```

   where:

   - *Java_home* is the location of the compatible version of Java.
   - *jarFile* is the BRM installer file. For example:

     ```
     brmserver_15.2.0.0.0_linux_generic_full.jar
     ```

   - *Inventory_home* is the location of the Oracle inventory.
   - *parameter* is the name of an installation parameter.
   - *value* is the value of the installation parameter.
   - *path* is the absolute path to the response file.

   For example:

   ```
   $JAVA_HOME/bin/java -jar brmserver_15.2.0.0.0_linux_generic_full.jar -debug -
   invPtrLoc Inventory_home/oraInventory/oraInst.loc INSTALL_TYPE=Complete -
   responseFile /tmp/brm_complete.rsp  -silent
   ```

   The installation runs silently in the background.

> **Note**
>
> For a complete list of the available parameters for the [*parameter*=*value*] clause, and for information about optional debugging parameters, run the following command:
>
> ```
> Java_home/bin/java -jar jarFile -help
> ```

Upgrading the BRM Database Schema to the New Release

The database objects associated with the optional components that are not already installed are not created or updated. To upgrade the BRM database schema, perform one of these procedures:

- Upgrading the Schema on Single-Schema Systems
- Upgrading the Schema on Multischema Systems

> **Note**
>
> If you use a separate Invoice database schema, you must also upgrade that database schema. Apply any upgrade scripts applied to the main BRM database schema to the Invoice database schema as well.

Upgrading the Schema on Single-Schema Systems

To upgrade the schema on single-schema systems:

1. Grant an additional permission to the BRM database user:
   a. Connect to the primary database instance using SQL*Plus:

      ```
      % sqlplus system@databaseAlias as sysdba
      ```

   b. Run the following command:

      ```
      SQL> GRANT SELECT ON V_$SESSION TO BRM_user;
      Grant succeeded.
      SQL> commit;
      Commit complete.
      ```

   c. Exit SQL*Plus.
2. Open the *BRM_home*/**setup/scripts/pin_tables.values** file in a text editor.
3. Set these parameters to **local**:

   ```
   # For indexes on event tables
       $PIN_CONF_PARTITION_IND            = "local";
   # For indexes on non event tables
       $PIN_CONF_NON_EVENT_PARTITION_IND  = "local";
   ```

4. Save and close the file.
5. Go to *BRM_home* and source the **source.me** file:
   - Bash shell:

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

- C shell:

```
source source.me.csh
```

6. Go to the *BRM_home*/**setup/scripts** directory and run this script:

   > **Note**
   >
   > Ensure that you do not run the upgrade script in the background.

   ```
   perl script
   ```

   where *script* is either `pin_upgrade_15.pl` for BRM 15.2.0 or `pin_upgrade_15psx.pl` for a maintenance release with *x* as the maintenance release, such as 1 for BRM 15.2.1, 2 for BRM 15.2.2, and so on.

   This script runs a series of scripts that upgrade the BRM data to the new release.

7. 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:

1. On the primary BRM installation machine, do this:
   a. Grant an additional permission to the BRM database user:
      i. Connect to the primary database instance using SQL*Plus:

      ```
      % sqlplus system@databaseAlias as sysdba
      ```

      ii. Run the following command:

      ```
      SQL> GRANT SELECT ON V_$SESSION TO BRM_user;
      Grant succeeded.
      SQL> commit;
      Commit complete.
      ```

      iii. Exit SQL*Plus.
   b. Ensure that the Data Managers (DMs) in your secondary BRM installation machines are up and running.
   c. Ensure that the PERL_VERSION environment variable is set to the latest version of Perl certified with BRM.
   d. Open the *BRM_home*/**setup/pin_setup.values** file in a text editor.
   e. Set these parameters:

   ```
   $SETUP_DROP_ALL_TABLES = "NO";
   $SETUP_INIT_DB = "YES";
   $SETUP_CONFIGURE = "YES";
   $SETUP_DATABASE_TABLES = "NO";
   ```

   f. Save and close the file.
   g. Open the *BRM_home*/**setup/scripts/pin_tables.values** file in a text editor.

h. Set these parameters to **local**:

```
# For indexes on event tables
    $PIN_CONF_PARTITION_IND             = "local";
# For indexes on non event tables
    $PIN_CONF_NON_EVENT_PARTITION_IND   = "local";
```

i. Save and close the file.

j. Go to *BRM_home* and source the **source.me** file:

   Bash shell:

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

   C shell:

```
source source.me.csh
```

k. Go to the *BRM_home***/apps/multi_db** directory and run this script:

```
./install.sh
```

   Follow the on-screen instructions, entering this 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 extended architecture (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.

l. Go to the *BRM_home***/setup/scripts** directory and run this script:

> **Note**
>
> Ensure that you do not run the upgrade script in the background.

```
perl script
```

where *script* is either **pin_upgrade_15.pl** for BRM 15.2.0 or **pin_upgrade_15ps***x***.pl** for a maintenance release with *x* as the maintenance release, such as 1 for BRM 15.2.1, 2 for BRM 15.2.2, and so on.

This script runs a series of scripts that upgrade the BRM data tot he new release.

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

n. Ensure that the BRM processes are not running.

2. On *each* secondary BRM installation machine, before upgrading, do this:

   a. Determine the next available index in the Oracle Wallet by running the following command:

   ```
   pin_config_editor -list -wallet $BRM_CONF_WALLET
   ```

   This should return a list of wallet entries like the following:

   ```
   oracle.security.client.connect_string1
   oracle.security.client.connect_string2
   oracle.security.client.password1
   oracle.security.client.password2
   oracle.security.client.username1
   oracle.security.client.username2
   ```

   In this case, the next available index would be **3**.

   b. Add the primary database to the wallet on the secondary machine using the following commands:

   ```
   pin_config_editor -setconf -wallet walletLocation -parameter
   oracle.security.client.usernameindex -value username
   pin_config_editor -setconf -wallet walletLocation -parameter
   oracle.security.client.passwordindex -value password
   pin_config_editor -setconf -wallet walletLocation -parameter
   oracle.security.client.connect_stringindex -value databaseAlias
   ```

   where:

   - *walletLocation* is the location of the wallet for the secondary BRM instance, for example **$PIN_HOME/wallet/client**.
   - *index* is the index you determined in the previous step.
   - *username* is the user name of the primary database.
   - *password* is the password for the database user on the primary database.
   - *databaseAlias* is the alias or connection string for the primary database.

   For example:

   ```
   pin_config_editor -setconf -wallet $PIN_HOME/wallet/client -parameter
   oracle.security.client.username3 -value pin01
   pin_config_editor -setconf -wallet $PIN_HOME/wallet/client -parameter
   oracle.security.client.password3 -value password
   pin_config_editor -setconf -wallet $PIN_HOME/wallet/client -parameter
   oracle.security.client.connect_string3 -value pindb01
   ```

   c. Grant an additional permission to the BRM database user:

      i. Connect to the primary database instance using SQL*Plus:

      ```
      sqlplus system@databaseAlias as sysdba
      ```

      ii. Run the following command:

```
SQL> GRANT SELECT ON V_$SESSION TO BRM_user;
Grant succeeded.
SQL> commit;
Commit complete.
```

iii. Exit SQL*Plus.

d. Ensure that the BRM processes are not running.

e. Open the *BRM_home***/setup/scripts/pin_tables.values** file in a text editor.

f. Set these parameters to **local**:

```
# For indexes on event tables
    $PIN_CONF_PARTITION_IND             = "local";
# For indexes on non event tables
    $PIN_CONF_NON_EVENT_PARTITION_IND   = "local";
```

g. Save and close the file.

h. Go to *BRM_home* and source the **source.me** file:

Bash shell:

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

C shell:

```
source source.me.csh
```

i. Copy the *BRM_home***/setup/scripts/pin_multidb.conf** file from the primary environment to the same location in the secondary environment.

> **Note**
>
> After you copy this file manually to the secondary environment, the upgrade script will refresh the views by reading the copied file.

j. Go to the *BRM_home***/setup/scripts** directory and run this script:

> **Note**
>
> Ensure that you do not run the upgrade script in the background.

```
perl script
```

where *script* is either **pin_upgrade_15.pl** for BRM 15.2.0 or **pin_upgrade_15ps*x*.pl** for a maintenance release with *x* as the maintenance release, such as 1 for BRM 15.2.1, 2 for BRM 15.2.2, and so on.

This script runs a series of scripts that upgrade the BRM data to the new release.

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

3. On the primary BRM installation machine, do this:

a. Go to the *BRM_home***/apps/multi_db** directory and open the **config_dist.conf** file in a text editor.

b. Add a line specifying the schema name at the end of each block in the file. For example, if you have two database schemas, your file might look like this:

```
DB_NO = "0.0.0.1" ;         # 1st database configuration block.
PRIORITY = 1 ;
MAX_ACCOUNT_SIZE = 100000 ;
STATUS = "OPEN" ;

DB_NO = "0.0.0.2" ;         # 2nd database configuration block.
PRIORITY = 1 ;
MAX_ACCOUNT_SIZE = 100000 ;
STATUS = "OPEN" ;
```

Update the file to look more like the following:

```
DB_NO = "0.0.0.1" ;         # 1st database configuration block.
PRIORITY = 1 ;
MAX_ACCOUNT_SIZE = 100000 ;
STATUS = "OPEN" ;
SCHEMA_NAME = "pindb01" ;

DB_NO = "0.0.0.2" ;         # 2nd database configuration block.
PRIORITY = 1 ;
MAX_ACCOUNT_SIZE = 100000 ;
STATUS = "OPEN" ;
SCHEMA_NAME = "pindb02" ;
```

using the correct schema names for your environment.

c. Save and exit the **config_dist.conf** file.

Upgrading the Pipeline Manager Database Schema to the New Release

> **Note**
>
> In multischema systems, run the database upgrade script on the primary schema and then on the secondary schemas.

To upgrade your Pipeline Manager database schema:

1. Open the *Pipeline_home***/upgrade/pipeline_upgrade.cfg** file in a text editor, where *Pipeline_home* is the directory in which Pipeline Manager is installed.
2. Set the values of the following parameters:
   - Set **PIN_TEMP_DIR** to the directory in which you want to create the temporary files.
   - Set the **$PIPELINE_TBLSPACE** environment variable to the tablespace where you want to create pipeline database objects.

- 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'} );
```

3. Save and close the file.
4. 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.
5. Go to the *Pipeline_home*/**setup/scripts** directory and run the following command:

   ```
   perl script
   ```

   where *script* is either **pin_upgrade_15.pl** for BRM 15.2.0 or **pin_upgrade_15ps*x*.pl** for a maintenance release with *x* as the maintenance release, such as 1 for BRM 15.2.1, 2 for BRM 15.2.2, and so on.

   This script runs a series of scripts that upgrade Pipeline Manager data to the new release.

Adding Customizations

> **Note**
>
> In multischema systems, incorporate customizations first on the secondary BRM installation machines and then on the primary BRM installation machine.

Incorporate any customizations you made, including source code, policy, **pin.conf**, **pin_ctl.conf**, **pin_setup.values**, and **Infranet.properties** files, if you have not already incorporated them.

(Production system only) Remove all entries for the **pin_virtual_time** utility from the configuration files.

Post-Upgrade Procedures for the New Release

After you install the new version of BRM, perform the following postinstallation tasks:

1. Create large indexes. See "Creating Large Indexes" for instructions.
2. Configure SSL for the BRM Database. See "Configuring SSL for the BRM Database" for instructions.
3. Remove invalid objects from the database. See "Removing Invalid Objects from the Database" for instructions.
4. If you intend to migrate legacy account information into BRM, run a database script to prepare. See "Preparing to Migrate Legacy Account Data" for instructions.

Creating Large Indexes

For performance reasons, you must create some indexes manually after the upgrade.

To create the indexes in the default tablespace, use the following commands:

```
CREATE INDEX i_purchased_discount_acc_obj__id ON purchased_discount_t
(account_obj_id0);
CREATE INDEX i_purchased_product_acc_obj__id ON purchased_product_t
(account_obj_id0);
CREATE UNIQUE INDEX i_billinfo_id_code__id ON billinfo_t
(identification_code);
```

To create the indexes in another tablespace (for example, the index tablespace), use the following commands:

```
CREATE INDEX i_purchased_discount_acc_obj__id ON purchased_discount_t
(account_obj_id0) tablespace tablespace_name;
CREATE INDEX i_purchased_product_acc_obj__id ON purchased_product_t
(account_obj_id0) tablespace tablespace_name;
CREATE UNIQUE INDEX i_billinfo_id_code__id ON billinfo_t
(identification_code) tablespace tablespace_name;
```

where *tablespace_name* is the name of the tablespace that should contain the indexes.

Installing Optional Components After the Release Upgrade

Typically, you install optional components before installing the new release. However, if you have already upgraded BRM and you want to install any optional components:

1. Set the PERL_HOME environment variable to the directory in which Perl is installed by running this command:

   ```
   setenv PERL_HOME Perl_path
   ```

   where *Perl_path* is the path to the directory in which Perl is installed, for example **/perl_5_36.0/linux**.

   For the compatible Perl version, see "Additional Software Requirements (BRM)" in *BRM Compatibility Matrix*.

2. Install the required optional components. See "Installing Optional Components".
3. Unset the PERL5LIB environment variable.
4. Go to the *BRM_home*/**setup** directory and run this command:

   ```
   ./pin_setup -GA
   ```

5. Upgrade the optional component to the new release. See "Upgrading to the New Release".
6. Set the PERL5LIB environment variable to the directory in which Perl is installed.

18

Rolling Back Your BRM Database Upgrade

Learn how to roll back your Oracle Communications Billing and Revenue Management (BRM) 15.2 database upgrade to its prior version.

Topics in this document:

- About Rolling Back Your BRM Database
- Tasks for the Database Rollback Process
- About Reusing Your Old BRM Installation

About Rolling Back Your BRM Database

The rollback feature allows you to revert your BRM database and other supported components to their previous stable state if issues occur after an upgrade. For example, if you upgrade your database from BRM 12.0.0.8.0 to BRM 15.2 and encounter problems, you can roll it back to BRM 12.0.0.8.0.

> **Note**
>
> You can roll back your BRM database only to the exact version that existed before the upgrade. For instance, if you upgrade from BRM 15.0 to 15.2, you can roll back only to 15.0. You cannot roll back to 15.1 or 12.0.0.*x*.0.

Before rolling back your BRM database, carefully review all prerequisites. Avoid making changes to system keys between the upgrade and rollback, and ensure you complete any rollbacks before making further schema or environment changes. In particular:

- Do not add secondary schemas to your BRM database before performing a rollback. Adding a schema first may hinder the rollback process.
- Do not rotate root keys between the upgrade and rollback.
- Do not add new optional managers to your existing BRM setup between the upgrade and rollback.
- Run the rollback script using BRM 15.2 (the current BRM version). After the database schema rollback completes successfully, revert to the previous file system and verify that the supported database server and client are in place before you restart services.

The BRM database rollback feature has the following limitations:

- Rolling back is supported only for BRM 15.2 or later. You cannot roll back from BRM 15.1 or earlier versions.
- Rolling back is limited to BRM 12.0.0.4.0 or later. You cannot revert to earlier releases.
- Rolling back can be performed only immediately after an upgrade.
- Rolling back is not supported for full BRM installations. To revert after a full installation, you must reinstall the earlier version of the software.

- After rolling back from BRM 15.2 to BRM 12.x, attempting to read BRM 15.2 events in BRM 12.x fails if the event version supported in BRM 12.x is lower than the version used in BRM 15.2. Because no automatic event downgrade occurs, you must manually convert the BRM 15.2 event format to the BRM 12.x event version before these events can be accessed in the rolled-back environment.

After rolling back your BRM database, you can:

- Start using the previous BRM release. See "About Reusing Your Old BRM Installation".
- In case you want to upgrade your BRM database again, you can upgrade to the same release you had rolled back from or a higher version.

  For example, if you roll back from BRM 15.2 to BRM 15.0, you can subsequently upgrade again to BRM 15.2 or later, but you cannot upgrade to BRM 15.1.

Tasks for the Database Rollback Process

The following lists the tasks required to roll back your BRM database to a previously installed release version.

1. Ensure the upgrade to BRM 15.2 completed successfully.

   If the upgrade failed, resolve all issues before proceeding. The rollback procedure cannot be used in the middle of the upgrade flow.
2. Stop all BRM daemons, processes, and managers.

   See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.
3. Go to the *BRM_home*/**setup/scripts** directory.
4. Run the BRM database rollback script:

   ```
   perl pin_rollback.pl
   ```
5. Adjust your environment to point to the previously installed release version:
   - Restore all third-party software to versions supported by the previous release.
   - Update your environment variables, including those for Perl and Java, to reference the appropriate supported versions.

   For supported software versions, see the BRM Compatibility Matrix for your version of the software.
6. Verify that the database rollback completed successfully. Check that the ROLLBACK_STATUS field in the BRM_PS_T table is set to **1**.
7. If rolling back a minor upgrade, such as from 15.2 to 15.1, restore the previous BRM environment's file system:

   ```
   cd BRM_home/brm_backup
   sh fileSystemRollback.sh
   ```
8. Restart all BRM services.

   > **Note**
   >
   > If rolling back a major upgrade, such as from BRM 15.2 to 12.0.0.*x*.0, restart BRM services using the previous version's file system. For example, if you rolled back from 15.2 to 12.0.0.7.0, use the 12.0.0.7.0 file system.

About Reusing Your Old BRM Installation

After successfully rolling back your BRM database, you can resume using your system. However, before launching BRM, creating accounts, or running billing, do the following:

- If you rolled back to 15.0 or 15.1: Go to the *BRM_home***/sys/data/config** directory, and run the **load_pin_notify** utility.
- If you rolled back to 12.0.0.*x*.0: Enter the 12.0.0.*x*.0 file system, go to the *BRM_home***/sys/data/config** directory, and run the **load_pin_notify** utility.

For information about the utility's syntax and parameters, see "load_pin_notify" in *BRM Managing Customers*.

19

Troubleshooting the BRM Upgrade

Learn how to solve problems that may occur in your Oracle Communications Billing and Revenue Management (BRM) system during or after an upgrade.

Topics in this document:

- Problem: A Wallet Error Occurred When Upgrading the Database
- Problem: Missing Table While Upgrading the Database

For more information, see "Reference Guide to BRM Error Codes" and "Resolving Problems in Your BRM System" in *BRM System Administrator's Guide*.

Problem: A Wallet Error Occurred When Upgrading the Database

When you run the **pin_upgrade** script to upgrade the database from BRM 12.0 to BRM 15.1 or later, you may see an error like the following:

```
WARNING : WALLET_FILE_STORAGE_T table is missing
EXISTING_WALLET_LOCATION Can not be empty
```

Possible Cause

This may happen if you run the database upgrade script from a different server than the one where you ran the installer for the upgrade.

Solution

To remove this error:

1. Open the *BRM_home***/setup/pin_setup.values** file in a text editor.
2. Locate the following line in the file

   ```
   #$EXISTING_WALLET_LOCATION = "NA" ;
   ```

3. Remove the comment from the line and replace **NA** with the path to your existing wallet (the one for the version you are upgrading from).

   For example if you are upgrading from BRM 12.0 and your BRM 12.0 wallet is located in **/pinuser/opt/portal/BRM/wallet/client**, update the line to read:

   ```
   $EXISTING_WALLET_LOCATION = "/pinuser/opt/portal/BRM/wallet/client/" ;
   ```

4. Save and exit the file.
5. Run the **pin_upgrade** script again.

Problem: Missing Table While Upgrading the Database

When you upgrade BRM 12.0 to BRM 15.0 or later versions, you may encounter the following error in the **dm_oracle** log file while running the BRM 15.*x* installation script:

```
ORA-12514, TNS:listener does not currently know of service requested in connect
descriptor
```

Following this, when you run the **pin_upgrade** script to upgrade the database, the below error appears in the **dm_oracle** log file:

```
ORACLE error: load_brm_root_wallet: PINStmtExecute: code 942, op 0
=ORA-00942: table or view does not exist
```

This indicates that the BRM 15.*x* installation script failed to create the WALLET_FILE_STORAGE_T table. This table is essential for the **pin_upgrade** script to create the root wallet.

Possible Cause

The table creation gets skipped due to a connectivity issue between the BRM server and the database.

Solution

To remove this error:

1. Check the network connectivity between the designated BRM server and the database server before running the BRM 15.*x* installation script. This can be validated using the ping command.
2. Ensure that **tnsping**, **sqlplus**, and **jdbc** connectivity is working correctly.
3. Edit the **sqlnet.ora** file for the target database to increase the amount of time specified for the BRM server to connect with the database server and provide the necessary authentication information. If no units are specified, the value is in seconds. For example, to set the timeout to 120 seconds, update the parameter this way:

   ```
   SQLNET.INBOUND_CONNECT_TIMEOUT=120
   ```

   The default value is **3**.
4. Rerun the BRM 15.*x* installation script.
5. Verify that the WALLET_FILE_STORAGE_T table is present and contains the required data.
6. Proceed with upgrading the database by running the **pin_upgrade** script.

For more information, see "Upgrading BRM 12.0 to BRM 15.2" .