

Oracle® Application Integration Architecture

Oracle Communications Billing and Revenue Management
Integration Pack for Oracle E-Business Suite: Revenue
Accounting Implementation Guide

Release 11.1

E27226-01

January 2012

Documentation that describes the process integration for revenue management, which is part of the Oracle Communications Billing and Revenue Management Integration for Oracle E-Business Suite: Revenue Accounting. The process integration for revenue management moves general ledger (GL) data from Oracle Communications Billing and Revenue Management (Oracle BRM) to the Oracle E-Business Suite (Oracle EBS), enabling customers to use Oracle EBS General Ledger (Oracle GL) as an accounting engine on top of the Oracle BRM application.

Copyright © 2001, 2012, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

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

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

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

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

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

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

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	xi
What's New in this Guide	xi
Common Oracle AIA Pre-Built Integration Guides	xii
Documentation Accessibility	xii
Additional Resources	xii
 1 Understanding the Process Integration for Revenue Management	
1.1 Revenue Management Overview	1-1
1.2 Key Benefits	1-2
1.3 Participating Applications	1-2
1.3.1 Oracle Communications BRM	1-2
1.3.2 Oracle E-Business Suite Financials - General Ledger	1-2
1.4 Revenue Management Business Process Flow	1-3
1.5 Solution Assumptions and Constraints	1-4
1.5.1 Solution Assumptions	1-4
1.5.2 Solution Constraints	1-6
 2 Setting Up the Process Integration for Revenue Management	
2.1 Prerequisites	2-1
2.2 Setting Up the Revenue Management Integration	2-2
2.3 Performing Setup Tasks	2-3
2.3.1 Setup Tasks for Oracle GL	2-3
2.3.2 Setup Tasks for Oracle BRM	2-3
2.3.3 Setup Task for the Integration Layer	2-4
2.4 Configuring and Generating GL Data in Oracle BRM	2-4
2.4.1 How to Configure Data in Oracle BRM	2-4
2.4.2 How to Generate Data in Oracle BRM	2-5
2.5 Picking Up and Transforming the GL Data	2-5
 3 Configuring and Running the Process Integration for Revenue Management	
3.1 Changing the Oracle GL Database Schema Name	3-1
3.2 Setting Up a Schedule	3-6
3.3 Configuring the Process Integration for Revenue Management	3-7
3.4 Running the Process Integration for Revenue Management	3-9
3.4.1 Starting the Scheduler Agent	3-10

3.4.2	Stopping the Scheduler Agent	3-10
-------	------------------------------------	------

A Oracle BRM GL Configuration File

B Oracle BRM XML GL Report

C Oracle BRM GL Report

List of Figures

1-1	Revenue Management Business Process Flow	1-4
2-1	Revenue Management Load Process	2-6
3-1	Opening Topology Manager Example.....	3-2
3-2	ODI - Design Studio.....	3-2
3-3	Oracle Data Integrator Login	3-2
3-4	Topology Manager - Topology	3-3
3-5	Topology Manager - Topology - Physical Architecture	3-4
3-6	Physical Schema	3-4
3-7	Topology Manager - Topology - Physical Architecture	3-5
3-8	Physical Schema	3-6
C-1	Sample Oracle BRM GL Report	C-1

List of Examples

A-1	Sample Oracle BRM GL configuration file.....	A-1
B-1	Sample Oracle BRM GL report	B-1

List of Tables

1-1	Revenue Management: Overall Process	1-4
3-1	Value Attributes	3-8
3-2	Value Attributes: Filtering Account Types	3-9

Preface

Welcome to the Oracle Communications Billing and Revenue Management Integration Pack for Oracle E-Business Suite: Revenue Accounting Implementation Guide.

What's New in this Guide

- The Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations is restructured into a general installation chapter with an individual configuration and deployment chapter for each pre-built integration.
- The implementation guides are restructured into two parts: design and set up.
 - Part I - Design: This part provides functional overviews, activity diagrams, assumptions and constraints, and technical sequence diagrams and steps.
 - Part II - Set up: This part provides prerequisites, data requirements, and configuration steps.

What's New for Oracle Communications Pre-Built Integrations Product Documentation

- Oracle AIA for Communications release 11.2 includes these integrations to automate the Order to Cash process for the Communications industry.
 - Oracle Communications Order to Cash Integration for Siebel CRM
 - Oracle Communications Order to Cash Integration for Oracle Order and Service Management
 - Oracle Communications Order to Cash Integration for Oracle Billing and Revenue Management
- The Oracle Communications Order to Cash pre-built integration combines the functionality and replaces *Siebel CRM for Oracle Communications Billing and Revenue Management: Order to Bill* and *Oracle Order to Activate for Siebel CRM and Oracle Communications Order and Service Management* integration packs.

The key benefits of combining these integrations are:

 - Eliminates confusion on the differences between Order to Activate and Order to Bill integrations.
 - Provides additional flexibility for customers to choose and license components they really need.
- No new features have been added or removed.

- Oracle AIA Communications pre-built integrations 11.2 includes the following implementation guides:
 - Oracle Communications Order to Cash for Siebel CRM, Oracle Order and Service Management, and Oracle Billing and Revenue Management 11.2
 - Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care 11.2
 - Oracle Communications Billing and Revenue Management Integration Pack for Oracle Business Suite: Revenue Accounting 11.2

Common Oracle AIA Pre-Built Integration Guides

Oracle Application Integration Architecture Pre-Built Integrations 11.2 includes the following guides shared by all products delivered with this release:

- Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations Release 11.2

This guide provides an overview of the installation process, including how to install, configure, and deploy your pre-built integrations. The steps required to upgrade your pre-built integrations to the latest release are also provided.

- Oracle Application Integration Architecture Pre-Built Integrations 11.2: Utilities Guide

This guide describes:

- How to work with and configure Session Pool Manager (SPM), which is a service in the Oracle SOA Suite web server whose primary function is to manage a pool of web server session tokens that can be reused by BPEL flows.
- How to deploy and configure the AIACompositeScheduler. This is a utility component that is used by pre-built integrations to schedule a service-oriented architecture (SOA) composite to be invoked at the specified time interval.

- Oracle Application Integration Architecture Pre-Built Integrations 11.2: Product-to-Guide Index

The Product-to-Guide index lists the guides that provide information for each product delivered in this release.

Documentation Accessibility

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

Access to Oracle Support

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

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

Additional Resources

The following resources are also available:

- **Oracle Application Integration Architecture Foundation Pack:**

Oracle AIA Pre-Built integrations require Foundation Pack 11.1.1.5.0 to be installed. Refer to the Foundation Pack documentation library on OTN to download the Foundation Pack guides at

http://download.oracle.com/docs/cd/E21764_01/aia.htm.

- **Oracle Application Integration Architecture: Product-to-Guide Index:**

Oracle Technology Network:

<http://www.oracle.com/technetwork/index.html>

- **Known Issues and Workarounds:**

My Oracle Support: <http://tinyurl.com/83q8grw>

- **Release Notes:**

Oracle Technology Network:

<http://www.oracle.com/technetwork/index.html>

- **Documentation updates:**

Oracle Technology Network:

<http://www.oracle.com/technetwork/index.html>

Understanding the Process Integration for Revenue Management

This chapter provides an overview of the process integration for revenue management and discusses its key benefits, participating applications, and the business process flow.

This chapter includes the following sections:

- [Section 1.1, "Revenue Management Overview"](#)
- [Section 1.2, "Key Benefits"](#)
- [Section 1.3, "Participating Applications"](#)
- [Section 1.4, "Revenue Management Business Process Flow"](#)
- [Section 1.5, "Solution Assumptions and Constraints"](#)

1.1 Revenue Management Overview

The process integration for revenue management is part of the Oracle Communications Billing and Revenue Management Integration for Oracle E-Business Suite: Revenue Accounting. The process integration for revenue management moves general ledger (GL) data from Oracle Communications Billing and Revenue Management (Oracle BRM) to the Oracle E-Business Suite (Oracle EBS), enabling customers to use Oracle EBS General Ledger (Oracle GL) as an accounting engine on top of the Oracle BRM application.

Oracle BRM is the master for the process integration for revenue management. A GL report generation utility in Oracle BRM, generates GL data. This utility can be started manually or automatically.

1. Oracle BRM produces XML-based files that contain a complete and formatted GL data set based on summary reports.
2. The Oracle Data Integrator (ODI) picks up the reports and transforms them into GL data based on the mapping (provided by the process integration for revenue management) of the GL XML elements to the table columns in the Oracle GL INTERFACE table.
3. ODI then loads the GL data into the Oracle GL INTERFACE table.

The Oracle GL INTERFACE table is where Journal Import receives accounting data that is imported from other systems into Oracle GL.

4. When Journal Import receives this data, it validates and converts the imported data into journal entries within Oracle GL.

The process integration for revenue management differs from other Oracle Communications integrations in that it does not use Enterprise Business Objects (EBOs) or other standard Oracle Application Integration Architecture (Oracle AIA) objects to complete this integration. Rather, the process integration for revenue management uses ODI to pick up the Oracle BRM GL data files and then move those files to the Oracle GL application.

This integration uses these XML files:

- The Oracle BRM XML data files are the result of generating the GL data in Oracle BRM.
- The AIAConfigurationProperties.xml file specifies the Set of Books (SOB) ID (with Oracle EBS R11.5.10 CU2) or Ledger ID (with Oracle EBS R12.1.1), Exchange Rate Type mappings for this integration, and other parameters required for this integration.
- The CurrencyCodeMapping domain value map (DVM) maps the Oracle BRM currency codes to their corresponding Oracle EBS values.

1.2 Key Benefits

This integration offers these key benefits:

- Strengthens revenue assurance with real-time charging and tighter controls between billing and financial applications.
- Enhances business information from integrated analytics and availability of accurate and consistent billing and financial data.
- Maintains regulatory compliances.

1.3 Participating Applications

Participating applications are:

- Oracle Communications BRM
- Oracle E-Business Suite Financials - General Ledger

1.3.1 Oracle Communications BRM

Revenue management is an end-to-end life cycle of processes for generating, capturing, and collecting revenue for each customer. Revenue management includes the ongoing process of analyzing, evaluating, and optimizing each phase of the life cycle, providing complete insight and intelligence into the revenue relationships that customers have with a service provider. Oracle Communications BRM provides a product-based solution for revenue management. It is built to industry standards on a highly available, real-time platform, and it is functionally rich enough to support all communications and media service provider business processes across all lines of business.

For more information, see the *Oracle Communications Billing and Revenue Management (BRM) Documentation*.

1.3.2 Oracle E-Business Suite Financials - General Ledger

Oracle GL is a comprehensive financial management solution that provides highly automated financial processing, effective management control, and real-time visibility

into financial results. It enables an organization to meet financial compliance and improve its bottom line. Oracle GL is part of the Oracle E-Business Suite.

In today's complex, global, and regulated environment, finance organizations face challenges in trying to obey local regulations and multiple reporting requirements. Oracle GL allows companies to meet these challenges in a streamlined and automated way. They can assign multiple ledgers to a legal entity to meet statutory, corporate, regulatory, and management reporting. They can maintain all accounting representations simultaneously for a single transaction. For example, a single journal entered in the main record-keeping ledger can be automatically represented in multiple ledgers even if each ledger uses a different chart of accounts, calendar, currency, and accounting principle.

Oracle E-Business Suite Release 11.5.10 CU2 uses SOB (Set of Books). Release 12 replaces the SOB with Subledger Accounting (SLA), Ledgers, and Ledger Sets. Important differences exist between SOBs and Ledgers. A SOB is a general ledger with associated subledger books (AR, AP, and so on), and is self-contained regarding management, calendaring, and reporting. In Release 12, the SLA tables provide formal accounting for the subledgers. Ledger is the general data-transactions and balances-for a particular accounting entity. A Ledger Set is where management, calendaring, and GL Reporting live. The new multiple ledger approach keeps the data segmented, similar to a SOB, but unlike the SOB approach allows for processing and reporting across different accounting entities. Therefore, a group with several companies in a country where regulatory compliance demands one ledger for each company can maintain distinct ledgers, but the group can handle them for close, reporting, and adjustment purposes as if they were one.

Ledgers in a Ledger Set must have a common chart of accounts (COA). A COA is defined as a single global COA shared across an enterprise. In addition to sharing a COA, ledgers must share a common calendar to be in a Ledger Set, and different currencies and accounting principles are acceptable. The organization can capture and report on any number of currencies from the balance-level to the subledger-level. Currency conversion, revaluation, re-measurement, and translation are all performed in accordance with local and international accounting standards to improve internal controls and increase efficiency.

For more information, see the *Oracle E-Business Suite Financials - General Ledger Guide*.

1.4 Revenue Management Business Process Flow

[Figure 1–1](#) illustrates the revenue management business process flow.

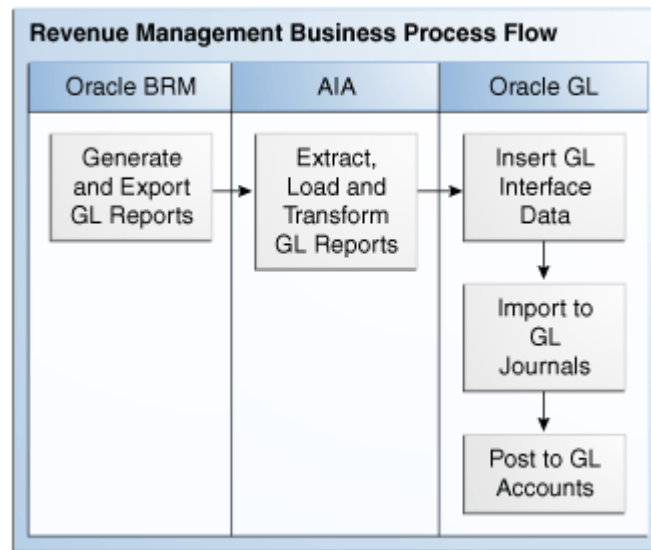
Figure 1–1 Revenue Management Business Process Flow

Table 1–1 illustrates the overall process for revenue management.

Table 1–1 Revenue Management: Overall Process

Work Location	Step
Oracle BRM	Step 1: Configure and generate GL data in Oracle BRM.
Integration process	<p>Step 2: The integration process picks up the XML files automatically or based on the schedule you set in ODI.</p> <p>Step 3: Transform Oracle BRM-generated XML data into the Oracle GL interface table.</p>
Oracle GL	Step 4: Import journal entries from the GL interface and then post them to GL accounts.

1.5 Solution Assumptions and Constraints

This section discusses the solution assumptions and constraints.

1.5.1 Solution Assumptions

The assumptions are:

1. Oracle GL accounts IDs are mapped and configured to Oracle BRM GL accounts.
2. In the scope of one type of report data, all reports for a single cycle are expected to be processed before the next cycle.
This action is recommended to avoid the possibility of picking up out-of-order reports.
3. This design addresses revenue, receivables, and liabilities.
4. If customers are required to use exact Oracle GL account Ids, then they must implement the import of Oracle GL COA into Oracle BRM as an implementation solution.

This implementation is not included in this release.

5. In the event of a configuration error in the Oracle BRM to Oracle GL account mapping, customers must manually reverse (N segments \times 7 types of reports \times M batches) and regenerate the feeds to Oracle GL.

N can be from tens to hundreds.

6. To address importing nonmonetary journals into Oracle GL (STAT currency), you must create and use a *dummy account*. This is because the extract from Oracle BRM GL Reports creates a Debit and Credit pair with the same amount in the Debit and the Credit.

Nonmonetary items increase or decrease in the debit credit or vice versa, depending on the setup, and Oracle GL does not require balanced journals to create journal entries for nonmonetary transactions.

However, for this integration you must create a balanced Journal Entry. As an example, let us say you are importing 100 free minutes into GL. The Billed Earned report extracts a Debit and a Credit of 100 free minutes. In Oracle GL, the way to set this up is to set up any GL account for this; one account must not be used for reporting purposes because it does not provide the actual net amount of 100 free minutes. So, let us say that you create account 7100 for the debit and 7200 (*dummy account*) for the credit with a classification of expense accounts. The net effect is 100 minutes, but the transaction creates a DR 7100 for 100 and Credit 7200 for 100. The reports must exclude account 7200 to report the 100 minutes. If you are required to reduce the amount to 80 free minutes, the corresponding entry is DR 7200 for 20 and CR 7100 for 20.

7. If the same report is loaded multiple times into the Oracle GL_Interface tables, it creates the additional accounting (double counting).

Therefore, if customers must reload a report into Oracle GL, they must reverse the original entries created by the first report run and then reload the report.

8. One instance of Oracle BRM can publish GL data to one Oracle E-Business Suite GL instance.

To target multiple SOB (with Oracle E-Business Suite R11.5.10 CU2) or Ledgers (with Oracle E-Business Suite R12.1.1) within a single GL instance, you must configure the Oracle AIA configuration file per source system and ensure that Oracle BRM publishes the report with the appropriate source system information.

The user must specify a SOB (with Oracle E-Business Suite R11.5.10 CU2) or a Ledger ID (with Oracle E-Business Suite R12.1.1) per source system.

9. Oracle BRM-generated GL reports are available in XML format.
10. Oracle BRM report output values for the four types of revenue accounts within a given GLID are *Gross*, *Net*, *Discount*, and *Tax*.

Choosing to use *Gross*, *Net*, or both is specific to the implementation. Implementations can also choose to ignore certain revenue account types even if valid GL accounts are populated. Therefore, the implementation must list in the integration layer all of the account types that must be ignored so that the process does not produce an error condition for those account types.

11. The time zone handling solution assumes that the Oracle BRM server and the Oracle Ebiz corporate time zone are set to the same value. The ODI flow that interfaces GL data from Oracle BRM to Oracle EBS GL does no time zone conversions.

1.5.2 Solution Constraints

This design assumes that the following statements are true:

1. Multiple Oracle BRM instances can contribute to GL data, but only one target GL system can exist.

Each XML file generated by any Oracle BRM instance can go to only one Oracle GL system. To support multiple Oracle GL systems, you must customize the integration so that each integration process instance targets a separate Oracle GL system. If you use multiple instances for each instance, then you must add a Source SystemID in the configuration file.

2. The Oracle BRM-generated XML file contains the Source SystemID from which it is generated.
3. One Oracle BRM instance publishes data to one SOB Id (with Oracle E-Business Suite R11.5.10 CU2) or to one Ledger Id (with Oracle E-Business Suite R12.1.1).

The same SOB Id or Ledger Id can be used by multiple Oracle BRM instances. Therefore, the relationship between an Oracle BRM instance and a SOB Id or Ledger Id is many to one.

4. Users must place the Oracle BRM-generated XML file in the defined integration folder.

If any of the processed XML files fail, users must manually place those files in the process folder for reprocessing after making the necessary corrections.

5. If the agent stops in the middle of processing of a file, and if that file is not in the success or the failure folder, then it is in the ODIInProgress folder.

Users must manually move this file from the ODIInProgress folder back to the input folder for reprocessing.

Setting Up the Process Integration for Revenue Management

This chapter describes how to set up the process integration for revenue management integration and lists the prerequisites. It discusses how to perform setup tasks, configure and generate general ledger (GL) data in Oracle Communications Billing and Revenue Management (Oracle BRM), and how to transform the GL data.

This chapter includes the following sections:

- [Section 2.2, "Setting Up the Revenue Management Integration"](#)
- [Section 2.1, "Prerequisites"](#)
- [Section 2.3, "Performing Setup Tasks"](#)
- [Section 2.4, "Configuring and Generating GL Data in Oracle BRM"](#)
- [Section 2.5, "Picking Up and Transforming the GL Data"](#)

2.1 Prerequisites

Complete these tasks before you install, deploy, configure, and run the process integration for revenue management:

1. Install ODI.
2. Install the revenue management integration.

For more information, see the *Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations*.

3. Install Oracle BRM GL and complete the initial setup tasks.

To obtain a complete list of the initial setup tasks, see the Oracle BRM product documentation.

4. Install Oracle E-Business Suite and Oracle General Ledger Release 11.5.10 CU2 or 12.1.1 and complete its initial setup tasks.

To obtain a complete list of Oracle GL setup tasks, refer to the Release 11.5.10 CU2 or 12.1.1 Oracle GL product documentation. Install Oracle E-Business Suite General Ledger Release 11.5.10 CU2 Seed Patch 6012471 or Oracle E-Business Suite General Ledger Release 12.1.1 Seed Patch 8344349. If the patch is absent or not available, make the following changes manually in Oracle GL:

5. Add the Portal GL Journal Source.
6. Add these Journal categories:

- Unbilled unearned
- Unbilled earned
- Billed unearned
- Billed earned
- Prior billed earned
- Billed
- Unbilled

Note: If you are applying Seed Patch to Oracle E-Business Suite General Ledger Release 12.1.1, manually change the Journal category *Prior Billed earned* to *Prior billed earned* in Oracle GL.

Journal categories are case sensitive and must match the Oracle BRM configuration.

2.2 Setting Up the Revenue Management Integration

To set up the revenue management integration, perform these steps:

1. Configure Oracle General Ledger (Oracle GL) and its Accounting Flexfield.
2. Map and configure Oracle GL accounts into Oracle BRM.
3. Generate GL data in Oracle BRM.

Rerun reports if the system identifies errors. The system moves GL data to the directory that you specified in Oracle BRM and Oracle Data Integrator (ODI).

4. Use ODI to pick up the GL data and load the GL Interface table with the accounting transactions.

Handle re-runs. If an error occurs, you must fix the source system and export the XML files again.

5. Use Oracle GL to import the journals into the Oracle GL application.

Manually or automatically import the reports into journal entries.

Optionally, you can delete the interface data if the system identifies an error before importing the interface data.

Use Oracle GL to review and post or delete. With deletion, the system deletes the data from both the interface tables and the journals.

6. Post journals to GL accounts using the Oracle GL application.

Manually or automatically post the imported journal entries into the corresponding GL accounts.

Optionally, delete the journal entries if the system identifies an error after the batch is imported but before it is posted.

Optionally, reverse the journal if the system identifies an error after posting.

7. Reconcile or review journal entries.

Review journal entries and, if errors occurred, create the appropriate adjustment entries.

2.3 Performing Setup Tasks

This section discusses these topics:

- [Section 2.3.1, "Setup Tasks for Oracle GL"](#)
- [Section 2.3.2, "Setup Tasks for Oracle BRM"](#)
- [Section 2.3.3, "Setup Task for the Integration Layer"](#)

2.3.1 Setup Tasks for Oracle GL

Complete these tasks in Oracle GL:

1. Define your Set of Books (SOB) with Oracle E-Business Suite R11.5.10 CU2 or ledgers with Oracle E-Business Suite R12.1.1.
2. Load the journal entry source seed data.

You can also specify whether you want Oracle GL to store journal reference information from Oracle BRM for a particular source.

Seed data is provided using a patch. Refer to the prerequisites section of this document for the patch number or manual steps if the seed patch is not available.
3. Load the journal entry categories seed data.

Seed data is provided using a patch. Refer to the prerequisites section of this document for the patch number or manual steps if the seed patch is not available.
4. If you want Journal Import to assign sequential numbers to your journal entries, enable sequential numbering, specifying *Automatic* as both your numbering method and document generation method.
5. Run the Optimizer program to create indexes for your account segments.
6. Define the concurrent program controls to improve the performance of Journal Import by setting the amount of disk space and memory it uses.

The Journal Import program requires approximately 1.4 megabytes of memory to run. You can also specify whether to save your Journal Import data each time you run Journal Import. Journal Import runs faster if you do not archive your data.
7. Disable dynamic insertion. Journal Import runs much faster when it is not required to create new account combinations dynamically.
8. Set the status of the accounting period (for which the integration is running) to either *Open* or *Future Enterable*.

2.3.2 Setup Tasks for Oracle BRM

Complete these tasks in Oracle BRM:

1. Configure the Oracle GL export configuration file.

In addition to the standard configuration parameters, you must also specify the Source System Identifier, which is the value of the SOB Id (with Oracle E-Business Suite R11.5.10 CU2) or Ledger Id (with Oracle E-Business Suite R12.1.1) mapping in the integration layer configuration file.
2. Customize the policy opcode PCM OP GL POL EXPORT GL to map the Oracle BRM GL account names to the appropriate Oracle GL accounts.

This mapping is done by modifying the relevant fields in the generated ledger report object. The Oracle GL accounts must be the concatenation of the individual Oracle GL segment values separated by the *V* segment separator string. This segment separator string must be the same as the one configured in the integration layer. For example, the system can map an internal Oracle BRM GL account name such as *west.california* to an Oracle GL account *XXX/YYY/ZZZ* if the three Oracle GL segments have values of *XXX*, *YYY*, *ZZZ*, and the segment separator.

Note: Customizing the opcode is only required if the Oracle BRM GL account names are different from the Oracle GL account names (including the segment separator). If the Oracle BRM and Oracle GL account names are the same, then you are not required to transform the names using the policy opcode. Also, the segment separator string is not required to be *V*, but both the policy opcode and the integration layer configuration must use the same separator string.

For more information, see *Oracle Communications Billing and Revenue Management (BRM) Documentation*, "BRM Documentation," Billing Customers, Collecting General Ledger Data.

2.3.3 Setup Task for the Integration Layer

For the integration layer, you must configure the parameters in the integration configuration file, including the mapping of the Source System Identifier to the SOB Id (with Oracle E-Business Suite R11.5.10 CU2) or Ledger Id (with Oracle E-Business Suite R12.1.1).

By just using the *Gross*, *Net*, or certain revenue account types when the GL account configuration contains *--N/A--* for the output values, list the account types that the integration layer must ignore in order not to get an error.

For more information about configuring ODI, see the *Oracle Fusion Middleware Installation and Upgrade Guide for Oracle Application Integration Architecture Foundation Pack*.

2.4 Configuring and Generating GL Data in Oracle BRM

This section discusses these topics:

- [Section 2.4.1, "How to Configure Data in Oracle BRM"](#)
- [Section 2.4.2, "How to Generate Data in Oracle BRM"](#)

2.4.1 How to Configure Data in Oracle BRM

Before running any GL data, you must set up Oracle BRM to:

- List all Oracle BRM GL accounts.
- Create GL Ids that assign GL codes to Oracle BRM balance impacts.
- Configure the summarization of Oracle BRM GL accounts into the Oracle GL accounts.

You can map Oracle BRM revenue into Oracle GL accounts in two ways:

- Configure the Oracle BRM GL accounts independently of the Oracle GL accounts, and map the Oracle GL accounts within the policy opcode.

- Configure the Oracle GL accounts within Oracle BRM GL configuration files.

If you cannot configure the exact name of an Oracle GL account, then you must customize Oracle BRM to configure the proper account Id. No pattern for configuring the accounts is specified, but typically Oracle BRM GL reporting is more detailed than Oracle GL, where accumulation is done.

Each GL segment contains its own set of GL Ids (and accounts). The balance impacts always use the GL Ids that are dependent on the product rated against, and they are independent of the GL segment with which the account is associated. The segments do not necessarily match the Oracle GL segments, but they should to simplify the integration.

For more information, see *Oracle Communications Billing and Revenue Management (BRM) Documentation*, "BRM Documentation," Billing Customers, Collecting General Ledger Data, About Collecting General Ledger data.

2.4.2 How to Generate Data in Oracle BRM

Oracle BRM can generate summary and detailed reports. You can use the summary report for this integration to Oracle GL. You can start with detailed report data and customize it to summarize the data, but what is handed to the integration is always a summary report.

You can pick a segment for reporting or, by default, all configured segments are reported.

In Oracle BRM, you must configure the GL account Ids to correspond with the target accounts in Oracle GL.

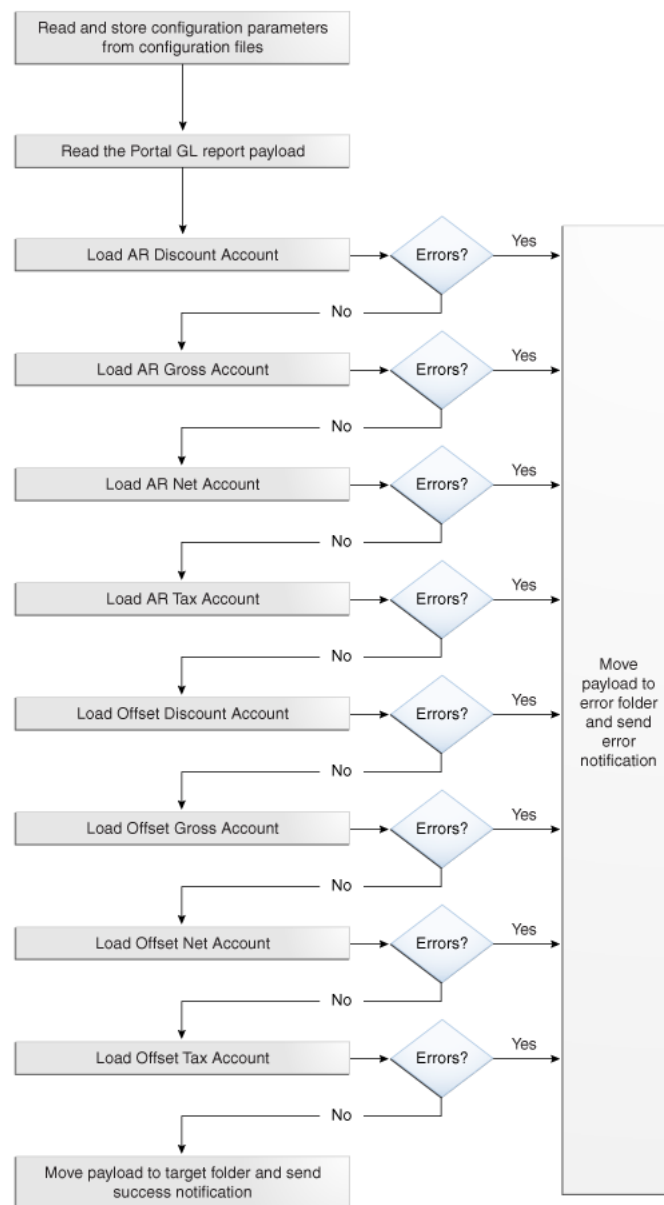
For more information, see *Oracle Communications Billing and Revenue Management (BRM) Documentation*, "BRM Documentation," Billing Customers, Collecting general ledger data, Loading general ledger configuration data. See also "BRM Documentation," Billing Customers, Collecting general ledger data, Exporting general ledger (G/L) reports to XML files.

2.5 Picking Up and Transforming the GL Data

The process integration for revenue management uses ODI to pick up the Oracle BRM data files, transform them from XML, and load the GL interface table. After the interface table is loaded, you use Oracle GL to import and post the journals.

In Oracle BRM, you configure the OutputDirectory to be the same directory that you specified in the AIAConfigurationProperties.xml file. After generating an Oracle BRM GL data file, when the file enters that directory, ODI picks it up and transforms it to the Oracle GL interface table.

[Figure 2-1](#) illustrates the load process:

Figure 2–1 Revenue Management Load Process

When this business process initiates, it:

1. Reads and stores configuration parameters from the configuration files.
2. Reads the Oracle BRM XML file.
3. Checks for the value of the property TargetID from AIAConfigurationProperties and executes the mapping for either Oracle E-Business Suite Release 11.5.10 CU2 or 12.1.1.
4. Checks to see if the AR discount account element is configured to be filtered out.
If it is, then the system moves to the next element. Otherwise, it loads the AR discount account element.

If the load is successful, the system moves to the next element.

If the load fails, the system moves the Oracle BRM XML file to the error folder, stops processing, and sends an email notification that the load failed.

5. Checks to see if the AR gross account element is configured to be filtered out.

If it is, then the system moves to the next element. Otherwise, it loads the AR gross account element.

If the load is successful, the system moves to the next element.

If the load fails, the system moves the Oracle BRM XML file to the error folder, stops processing, and sends an email notification that the load failed.

6. Checks to see if the AR net account element is configured to be filtered out.

If it is, then the system moves to the next element. Otherwise, it loads the AR net account element.

If the load is successful, the system moves to the next element.

If the load fails, the system moves the Oracle BRM XML file to the error folder, stops processing, and sends an email notification that the load failed.

7. Checks to see if the AR tax account element is configured to be filtered out.

If it is, then the system moves to the next element. Otherwise, it loads the AR tax account element.

If the load is successful, the system moves to the next element.

If the load fails, the system moves the Oracle BRM XML file to the error folder, stops processing, and sends an email notification that the load failed.

8. Checks to see if the offset discount account element is configured to be filtered out.

If it is, then the system moves to the next element. Otherwise, it loads the offset discount account element.

If the load is successful, the system moves to the next element.

If the load fails, the system moves the Oracle BRM XML file to the error folder, stops processing, and sends an email notification that the load failed.

9. Checks to see if the offset gross account element is configured to be filtered out.

If it is, then the system moves to the next element. Otherwise, it loads the offset gross account element.

If the load is successful, the system moves to the next element.

If the load fails, the system moves the Oracle BRM XML file to the error folder, stops processing, and sends an email notification that the load failed.

10. Checks to see if the offset net account element is configured to be filtered out.

If it is, then the system moves to the next element. Otherwise, it loads the offset net account element.

If the load is successful, the system moves to the next element.

If the load fails, the system moves the Oracle BRM XML file to the error folder, stops processing, and sends an email notification that the load failed.

11. Checks to see if the offset tax account element is configured to be filtered out.

If it is, then the system moves to the next element. Otherwise, it loads the offset tax account element.

If the load is successful, the system moves to the final step.

If the load fails, the system moves the Oracle BRM XML file to the error folder, stops processing, and sends an email notification that the load failed.

- 12.** Finally, the system moves the successful payload to the target folder and sends an email notification that the load is successful.

Configuring and Running the Process Integration for Revenue Management

This chapter discusses how to change the database schema name of Oracle E-Business Suite General Ledger (Oracle GL), set up a schedule, and configure and run the process integration for revenue management.

This chapter contains the following sections:

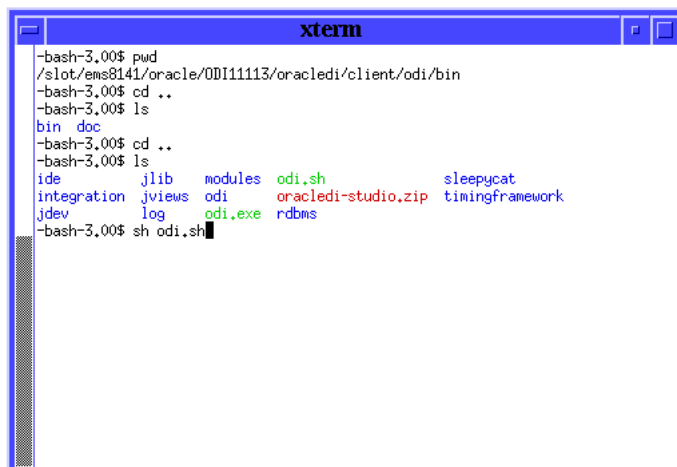
- [Section 3.1, "Changing the Oracle GL Database Schema Name"](#)
- [Section 3.2, "Setting Up a Schedule"](#)
- [Section 3.3, "Configuring the Process Integration for Revenue Management"](#)
- [Section 3.4, "Running the Process Integration for Revenue Management"](#)

3.1 Changing the Oracle GL Database Schema Name

The standard Oracle Application Integration Architecture (Oracle AIA) installation process assumes the Oracle GL database schema name to be the same as the database user name. If not, then you must run the following steps post installation.

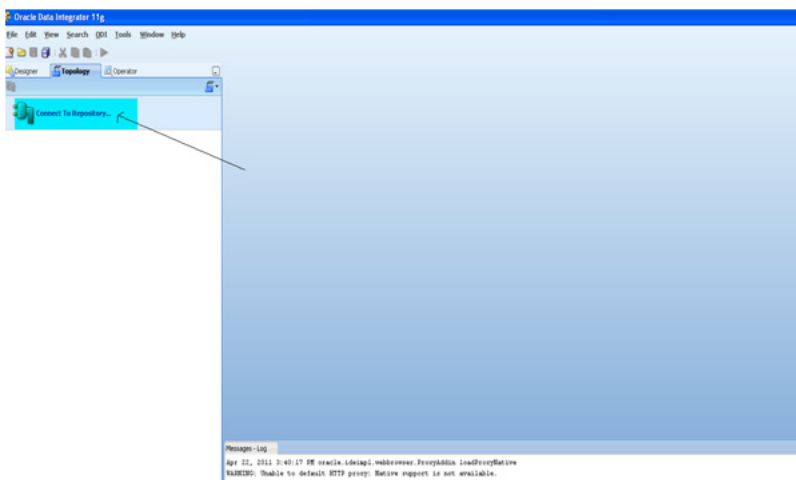
To change the Oracle GL database schema name:

1. Open a command prompt.
2. Source the file `aiaenv.sh` (if UNIX) or `aiaenv.bat` (if Windows) located in `<AIA_HOME>/aia_instances/AIAHOME/bin`.
3. Navigate to the directory `<ODI_HOME>/oracledi/client`.
4. Open Topology Manager, as shown in [Figure 3-1](#), by running the following command:
 - **UNIX:** `./odi.sh`
 - **Windows:** `odi.exe`

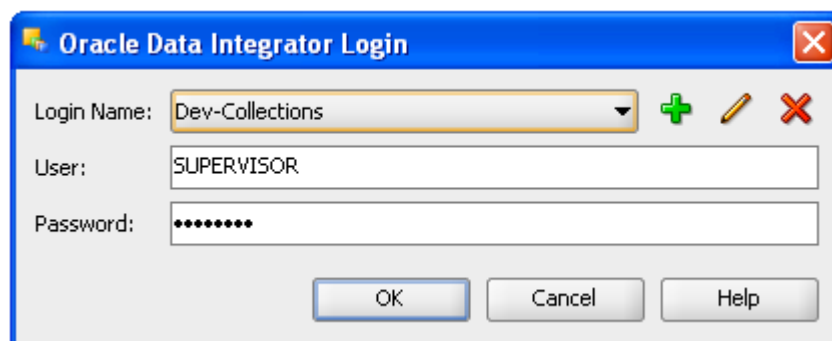
Figure 3–1 Opening Topology Manager Example

```
-bash-3.00$ pwd
/slot/ems8141/oracle/ODI111113/oracledi/client/odi/bin
-bash-3.00$ cd ..
-bash-3.00$ ls
bin doc
-bash-3.00$ cd ..
-bash-3.00$ ls
ide      jlib      modules  odi.sh      sleepycat
integration jviews   odi      oracledi-studio.zip timingframework
jdev     log      odi.exe  rdbms
-bash-3.00$ sh odi.sh
```

The Oracle Data Integrator (ODI) design studio appears, as shown in [Figure 3–2](#).

Figure 3–2 ODI - Design Studio

- Click **Connect to Repository** to open the ODI Login, as shown in [Figure 3–3](#).

Figure 3–3 Oracle Data Integrator Login

- Select *Repository* in the **Login Name** field.

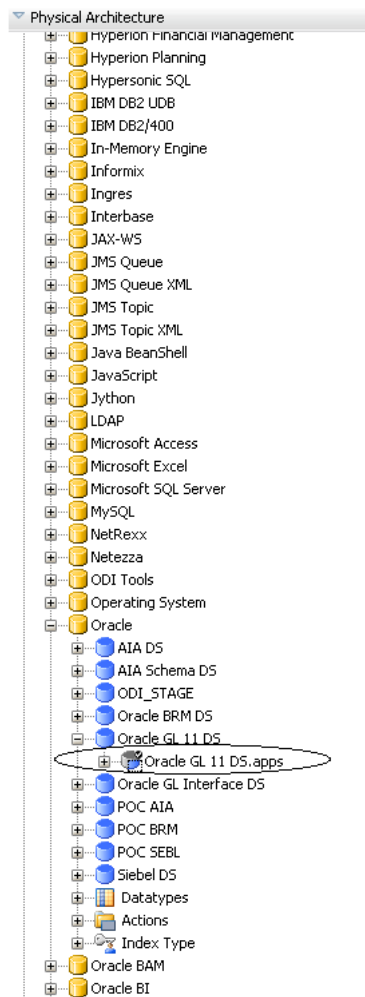
7. Enter the user name and password in the identified fields.
8. Click **OK** to open the Topology Manager.
9. Go to the **Topology** tab, as shown in [Figure 3-4](#).

Figure 3-4 Topology Manager - Topology



10. Navigate to the Physical Architecture, if you are not there.
Expand **Technologies > Oracle**.
11. If the version of Oracle GL is 11.5.10, expand Oracle GL 11 DS, as shown in [Figure 3-5](#).

Figure 3–5 Topology Manager - Topology - Physical Architecture



12. Right-click the child element, and click **Open**. (If the version is 12.1.1, see step 16.)

The Physical Schema screen appears, as shown in [Figure 3–6](#).

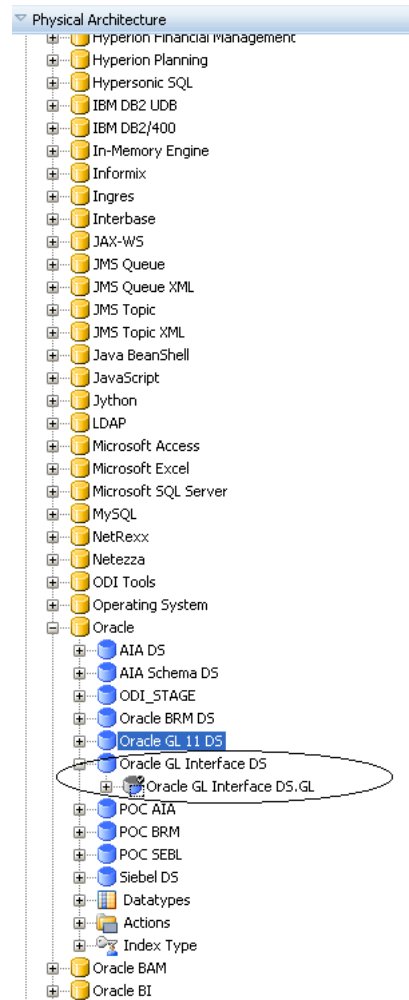
Figure 3–6 Physical Schema

13. Change the values in the **Schema (Schema)** and **Schema (Work Schema)** fields to the actual GL database schema name.

Note: Both the **Schema (Schema)** and the **Schema (Work Schema)** fields must contain the same value.

14. Click the **Apply** button.
15. Click the **OK** button.
16. If the version of Oracle GL is 12.1.1, expand Oracle GL Interface DS, right-click the child element, and click **Open**, as shown in [Figure 3–7](#).

Figure 3–7 Topology Manager - Topology - Physical Architecture



17. In the Physical Schema window, change the values of **Schema (Schema)** and **Schema (Work Schema)** to the actual GL database schema name, as shown in [Figure 3–8](#).

Note: Both the **Schema (Schema)** and the **Schema (Work Schema)** fields must contain the same value.

Figure 3–8 Physical Schema

The screenshot shows the 'Physical Schema' configuration window in Oracle Data Integrator. The title bar indicates it's for 'Data Server: Oracle GL Interface DS'. The 'Name' field is 'Oracle GL Interface DS-GL'. The 'Schema (Schema)' and 'Schema (Work Schema)' fields are both set to 'GL'. The 'Default' checkbox is checked. The 'Work Tables Prefix' section has fields for Errors, Loading, Integration, and Temporary Indexes. The 'Journalizing elements prefixes' section has fields for Databases, Views, and Triggers. The 'Naming Rules' section includes fields for Local Object Mask, Remote Object Mask, Partition Mask, Sub-Partition Mask, Local sequence mask, and Remote sequence mask.

18. When done, click **Apply**.
19. Click **OK**.
20. Regenerate the scenario `LOAD_PORTAL_DATA_TO_ORACLE_GL_PKG` Version 001 from the **Designer** tab.

3.2 Setting Up a Schedule

The standard Oracle AIA installation process completes the standard install for the process integration for revenue management. After the standard install, you must complete the following steps.

For more information, see the *Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations*.

1. Run the following command in the `oracledi/client` directory to open the Designer:
 - **UNIX:** `./odi.sh`
 - **Windows:** `odi.exe`
2. The Oracle Data Integrator (ODI) Login window appears. Enter the appropriate credential to log in to the Design Studio. If you have not set up the login, click the **New** button to create a login. The Work Repository Connection window appears.
3. Enter the following values:

Login Name: The name by which you identify this integration at login. For example, Revenue Management.

User: User name of the Designer, for example, SUPERVISOR.

Password: Password for the user, for example, SUNOPSIS.

User: The master repository user, for example, odimaster.

Password: The password for the master repository user, for example, odimaster.

Driver List: Select *Oracle JDBC Driver*.

Driver Name: Enter `oracle.jdbc.driver.OracleDriver`.

Url: Enter `jdbc:oracle:thin:@<host>:<port>:<sid>` where the host, port, and sid represent the database environment where you created the master repository.

Work Repository: Click the button to the right to display a list of available work repositories.

The system displays only one work repository. The repository name is the one that you chose during the standard installation.

Default Connection: Select this check box if you have multiple login names.

4. Click **OK**.

If the Work Repository Connection asks for a password, enter the Work Repository database user password that you set during the standard installation.

The Designer window appears.

5. Expand the **Portal to Oracle GL Project** tree to **Portal To Oracle GL Project, Packages, Load Portal Data To Oracle GL Pkg, Scenarios, LOAD PORTAL DATA TO ORACLE _GL_PKG Version 001, Scheduling**.

6. Double-click the existing RevMgmtAgent scheduler.

The Scenario Scheduling window appears.

7. Enter or select the following values:

Context: Select *Portal to GL Context*.

Agent: Select *RevMgmtAgent*.

Log Level: 5.

Status: Select the *Active* option.

Execution: Select the *On startup* option.

8. Click the **Execution Cycle** tab.

9. Enter or select the following values:

Repetition: Select *Many* times.

Interval between Repetitions: Enter 30 and choose *Second(s)*.

10. Click **OK**.

The scheduler is now configured.

3.3 Configuring the Process Integration for Revenue Management

After you install the process integration for revenue management, you must configure several parameters. Open the AIAConfigurationProperties.xml file that resides in the <aia_instances>/config/ folder. Entries in the AIAConfigurationProperties.xml file are case sensitive.

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Building AIA Integration Flows," How to Set Up AIA Workstation.

Configure the following parameters:

1. Under the moduleName RevMgmtSetOfBooksIDs find the <Property Name="Source SystemID ">202</Property> property tag.
 - Replace the tag value **SourceSystemID** with the value of the SourceSystemID that comes in the in Oracle BRM XML GL report. For example, if you have two Oracle BRM instances feeding into Oracle E-Business Suite General Ledger (Oracle EBS GL) and the report from one instance has the SourceSystemID value as *abc* and the report from the other instance has the value as *def*, then change the section as follows:

```

<ModuleConfiguration moduleName="RevMgmtSetOfBooksIDs">
  <Property name="abc">101</Property>
  <Property name="def">202</Property>
  .....
</ModuleConfiguration>

```

You must ensure that the value of the **SourceSystemID** in the Oracle BRM report is a value in the AIAConfigurationProperties.xml file.

- Replace the value of the **SourceSystemID**, such as 202, with the Set of Books ID as defined in E-Business Suite R11.5.10 CU2 or with the Ledger ID defined in E-Business Suite 12.1.1.
2. Under the moduleName RevMgmtExchangeRateType find the <Property name="ExchangeRateType">DAILY</Property> property tag.

The attribute DAILY represents the Exchange Rate Type. Do not change the name attribute value. Replace DAILY with the required exchange rate type.

The Exchange Rate Type handles currency conversions. For this release, Oracle provides a placeholder for the Exchange Rate Type. The Exchange Rate Type is passed to the Oracle GL application and is handled by Oracle GL when loading GL report data from the interface table to Oracle GL.

3. Under the moduleName RevMgmtParameters are several property tags with Name and Value attributes.

Do not edit the **Name** attribute value.

Table 3–1 lists value attributes.

Table 3–1 Value Attributes

Name Attribute Value	Property Tag Value
TargetID	Specify whether the target application is E-Business Suite Release 11.5.10 CU2 or 12.1.1. This property contains the value <i>EBS_01</i> for 12.1.1 and <i>EBS_02</i> for 11.5.10 CU2. The default value is <i>EBS_01</i> . The values entered are case sensitive.
ToMailAddress	Specify the To email address of the person to whom an email is sent, notifying them that the integration process is complete.
FromMailAddress	Specify the From email address for the person who performs the integration process.
TargetFileLocation	Specify the folder name where you want the process to move the XML file after it is successfully loaded.
ErrorFileTargetLocation	Specify the folder name where you want the process to move the XML file if a failure occurs.
SegmentSeparator	Specify how the Oracle BRM XML document separates in representing the Oracle GL segments.
PayloadSourceLocation	Specify the folder name to which the Oracle BRM-generated XML files are moved. The staging/ODIInProgress folder cannot be used as the PayloadSourceLocation.
Mail Server	Specify the mail server name.
RevMgmtHome	Specify the location of the RevenueManagementHome folder. Normally, this folder is located at <AIAHome>/DISHome/RevenueManagementHome.

4. Under the moduleName RevMgmtAccountFiltering specify what types of accounts you want to filter out (that is, to not load to the Oracle General Ledger interface table), as shown in [Table 3–2](#).

Possible values are *Yes* and *No*, and they are case sensitive.

These parameters enable you to filter certain account types. If the value is *No*, then they are not filtered out (they are loaded). If the parameter value is *Yes*, then the corresponding revenue account type entries are filtered out (they are not loaded).

Table 3–2 Value Attributes: Filtering Account Types

Name Attribute Value	Property Tag Value
FilterARGrossAccount	Specify a <i>No</i> value to load ARGrossAccount entries. Specify a <i>Yes</i> value to filter out this account.
FilterARDiscountAccount	Specify a <i>No</i> value to load FilterARDiscountAccount entries. Specify a <i>Yes</i> value to filter out this account.
FilterARNetAccount	Specify a <i>No</i> value to load ARNetAccount entries. Specify a <i>Yes</i> value to filter out this account.
FilterARTaxAccount	Specify a <i>No</i> value to load ARTaxAccount entries. Specify a <i>Yes</i> value to filter out this account.
FilterOffsetGrossAccount	Specify a <i>No</i> value to load OffsetGrossAccount entries. Specify a <i>Yes</i> value to filter out this account.
FilterOffsetDiscountAccount	Specify a <i>No</i> value to load OffsetDiscountAccount entries. Specify a <i>Yes</i> value to filter out this account.
FilterOffsetNetAccount	Specify a <i>No</i> value to load OffsetNetAccount entries. Specify a <i>Yes</i> value to filter out this account.
FilterOffsetTaxAccount	Specify a <i>No</i> value to load OffsetTaxAccount entries. Specify a <i>Yes</i> value to filter out this account.

Whenever you change the AIAConfigurationProperties.xml file, you must restart the agent to reflect your changes.

3.4 Running the Process Integration for Revenue Management

You can schedule the integration between Oracle BRM and the Oracle GL system to occur automatically. If the scheduler agent is kept running, then whenever Oracle BRM generates the report in XML format and writes the XML document into the specified directory, the integration process picks up the XML file, transforms it, and then transfers it to the interface table of the Oracle GL system.

Note: You can start or stop the scheduler agent whenever you want, which enables you to control the integration process. Therefore, to generate the reports in Oracle BRM in the last week of every month, then you can start the scheduler agent at the beginning of the last week and stop it when the week ends.

For more information, see [Section 3.2, "Setting Up a Schedule."](#)

3.4.1 Starting the Scheduler Agent

A scheduler agent, RevMgmtAgent, is delivered and is running on default port 20910. Navigate to the oracledi/agent/bin subdirectory and use this command to start it:

- **UNIX:** `./agent.sh -NAME=RevMgmtAgent`
- **Windows:** `agent.bat -NAME=RevMgmtAgent`

This command starts the scheduler agent, which in turn launches a scenario. The scenario waits for an Oracle BRM XML report file to arrive in the specified directory. As soon as the report file arrives, the scheduler agent transfers the file to the Oracle GL systems interface table. You can monitor the status of this integration process using the ODI Operator.

3.4.2 Stopping the Scheduler Agent

To stop the transfer of Oracle BRM XML reports to the interface table of the Oracle GL system, use this command:

- **UNIX:** `./agentstop.sh`
- **Windows:** `agentstop.bat`

Oracle BRM GL Configuration File

This appendix provides a sample Oracle Communications Billing and Revenue Management (Oracle BRM) General Ledger (GL) configuration file.

[Example A-1](#) is a sample of the Oracle BRM GL configuration file:

Example A-1 Sample Oracle BRM GL configuration file

```
#
#"(#)%Portal Version: pin_glid:
#";
#
General Ledger (G/L) ID Definitions
#
# Copyright (c) 1999 - 2006 Oracle. All rights reserved.
#
# This material is the confidential property of Oracle corporation
# or its licensors and may be used, reproduced, stored or
# transmitted only in accordance with a valid Oracle license or
# sublicense agreement.
#
# Use this file to define G/L IDs that correspond to your accounting
# methods.
#
# After defining the G/L IDs, run the 'load_pin_glid' utility to
# create the new config/glid storable object and load the G/L IDs
# into the Infranet database.
#
# You can have multiple pin_glid files, each defining the G/L IDs
# for some segments of your customer base. Use the "gl_segment"
# entry in this file to specify the segments for which these G/L IDs
# apply.
#
# For more information, see "About Collecting General Ledger
# Information" in the Infranet online documentation.
# gl_segment
#
# Name of the G/L segment described by this configuration file.
# You can define multiple GL segments in this file. All the gl
# segments that are defined shall have the same glids. You have to
# specify at least one gl_segment in this file
#
# You may have different glid files and each glid file may
# have multiple segments. You must ensure that the gl segment
# names specified in different glid files do not conflict. If this
# happens, the system shall overwrite the gl segment with the data
# of the last loaded glid file
```

```

#
# The value you specify should match the value in the
# PIN_FLD_GL_SEGMENT in the account objects for which you use
# this set of G/L IDs.
#
# GLIDs are assigned to accounts through the cm pin.conf parameter
# gl_segment and can be overridden by the policy opcode
# pcm_op_pol_cust_actginfo
# Syntax for G/L Segment Definitions
#
Use this syntax to define a gl_segment:
#
#gl_segment <segment_name> [no_roll-up]
#
#
# segment_name
#
# You specify the segment name. The segment name follows a URL-like
# notation and enables you to nest G/L segments. There is a root
# segment in Infranet and it is named ".". If your system has
# additional child segments then you define them as <root
# segment>.<child_segment>. You can have unlimited level of nesting.
# Segment names are case sensitive and cannot have spaces.
#
# The segment name . is used to identify the root segment. It is
# reserved and you should not change it
#
# no_roll-up
#
# The no_roll-up is used to indicate whether you want the data for
# the gl segment to roll up to its parent. If you do not want the
# data of this gl_segment to roll up to its parent then you must
# specify no_roll-up. If you want the roll-up then you must leave it
# blank. The root segment (.) does not roll up even if you
# specify blank. This is a special case.
#
An example is shown below :
#
# gl_segment .
# gl_segment .A (Child segment A, rolls up to . )
# gl_segment .A.A1 (Child segment of A, rolls up to A and to .)
# gl_segment .A.A2 no_roll-up (Child segment of A, does not roll up # to A )
# gl_segment .B no_roll-up (Child segment of . but does not
# roll up)
# gl_segment .C (Child segment of . rolls up to . )
#
# Based on the previous example, if you prepare the ledger report for
#
# . The data from .A, .A.A1 and .C is included
# .A The data of .A and .A.A1 is included (.A.A2 is not included)
# .A.A1 The data of .A.A1 is included (no childs of .A.A1 exist)
# .A.A2 The data of .A.A2 is included (no childs of .A.A2 exist)
# .B The data of .B is included (no childs of .B exist)
# .C The data of .C is included (no childs of .C exist)
#
# The default file has only the root segment specified in it

gl_segment .
# COA_ID
#

```

```

# Name of the Chart of Accounts (COA) ID that shall be used to
# validate the gl accounts specified under a GL ID
#
# This is an optional parameter. If you want the glids to be
# validated against a COA_ID then you must specify this parameter.
# If you do not want validation against a COA then leave it blank or
# do not specify the
# parameter.
# If specify this parameter and do not specify any value then it is
# as good as not specifying the parameter.
#
# If a COA_ID value is provided then the program checks if it has
# been created. Refer to the Infranet documentation on how
# to create COA's.
# Hence you must load the COA before loading the pin_glid files
#
# The default is no COA_ID
#
# You can specify only one COA_ID in one pin_glid file.
# The same COA_ID can be used in multiple pin_glid files if
# required.
# COA_ID


# Syntax for G/L ID Definitions
#
# Use this syntax to define a G/L ID:
#
#glid (
# id      <uint>
# descr   <str>
# taxcode <str>
# gl_acct [type] [attribute] [debit acct] [credit acct]
#)
#
# The following sections describe each of these fields.
# id
#
# An identification number you assign to a G/L ID.
#
# This ID can then be referenced by rates.
#
# GLID values has to be assigned according to the following rules:
# 1. GLID from 0 to 1000000 - are reserved for Portal use.
Use values greater than 1000000 when defining your own G/L IDs. The data type must
be "unit".
# 2. GLID 0 - is journalized, but it is excluded from GL reports.
# This value is considered as undefined and should not be present
# in production system.
# 3. GLID from 1 to 99 - are excluded from journalizing and GL
# reports.
# 4. GLID from 100 - are journalized and included in GL reports.
#
# The combined length of the "id" field and the "descr" field must
# be less than 255 characters.
# descr
#
A short description of the G/L ID.
#
# This description is displayed on the Pricing Tool menus.

```

```

#
# The combined length of the "id" field and the "descr" field must
# be less than 255 characters.
taxcode
#
# Type of product being purchased.
#
# This value must match the tax code used for the rates that use
# this G/L ID.
gl_acct
#
# An array that describes a pair of G/L accounts. Each G/L ID can
# have multiple gl_acct arrays. Each array contains the following
# fields:
#
# -----
# type - The conditions related to the billing and accounting
#        cycles under which this pair of G/L accounts should be
#        used. There are seven valid types:
#
#        1. billed           5. unbilled_earned
#        2. unbilled         6. unbilled_unearned
#        3. billed_earned    7. prev_billed_earned
#        4. billed_unearned
#
#        For more information, see "About G/L account pair types"
#        in the Infranet online documentation.
# -----
#attribute - The conditions related to what the amount of the
#             balance impact represents under which this
#             pair of G/L accounts is necessary. There are four
#             values:
#
#             1. gross       3. net
#             2. disc        4. tax
#
#             For more information, see "About G/L account
#             attributes" in the Infranet online documentation.
#
#debit acct - The name of one Infranet G/L account that
#             is affected by this G/L ID. For example, an A/R
#             account appears here for G/L IDs that refer to
#             fees charged to the end user. The impact on this
#             account is the opposite of the impact on the
#             credit acct.
#
# credit acct - The name of one Infranet G/L account that
#               is affected by this G/L ID. For example, an
#               revenue account appears here for G/L IDs that
#               refer to fees charged to the end user. The impact
#               on this account is the opposite of the impact
#               on the debit acct.
#
#Validation between the gl account and chart of account type.
#
#
#For a debit general ledger account
#
#gl attribute      |      Chart Of Account Type
#

```

```

#
# Gross | Asset
# Net | Asset
# Tax | Asset
# Disc | Expense or Revenue
#
#
#For a credit general ledger account
#
#gl attribute | Chart Of Account Type
#
#
# Gross | Expense or Revenue
# Net | Expense or Revenue
# Tax | Liability
# Disc | Asset
#
#
#Default G/L ID
glid ( id
0
descr undefined
gl_acct unbilled net undef.debit undef.credit
gl_acct billed net undef.debit undef.credit
# G/L ID for purchase rate
glid ( id
101
descr Purchase Fees
gl_acct billed gross purchase.debit purchase.credit gl_acct
billed net purchase.debit purchase.credit gl_acct billed disc
purchase.credit purchase.debit gl_acct billed_earned gross purchase.debit
purchase.credit gl_acct billed_earned net purchase.debit purchase.credit gl_acct
billed_earned disc purchase.credit purchase.debit gl_acct unbilled gross
purchase.debit purchase.credit gl_acct unbilled net purchase.debit
purchase.credit gl_acct unbilled disc purchase.credit purchase.debit gl_acct
unbilled_earned gross purchase.debit purchase.credit gl_acct unbilled_earned net
purchase.debit purchase.credit gl_acct unbilled_earned disc purchase.credit
purchase.debit
# G/L ID for cycle_forward rate
glid ( id
102
descr Monthly Fees
gl_acct billed gross monthly.debit monthly.credit gl_acct
billed net monthly.debit monthly.credit gl_acct billed disc
monthly.credit monthly.debit gl_acct billed_earned gross
monthly.debit monthly.credit gl_acct billed_earned net
monthly.debit monthly.credit gl_acct billed_earned disc
monthly.credit monthly.debit gl_acct prev_billed_earned gross
monthly.debit monthly.credit gl_acct prev_billed_earned net
monthly.debit monthly.credit gl_acct prev_billed_earned disc
monthly.credit monthly.debit gl_acct billed_unearned gross
monthly.debit monthly.credit gl_acct billed_unearned net
monthly.debit monthly.credit gl_acct billed_unearned disc
monthly.credit monthly.debit gl_acct unbilled gross
monthly.debit
monthly.credit gl_acct unbilled net
monthly.debit monthly.credit gl_acct unbilled disc
monthly.credit monthly.debit gl_acct unbilled_earned gross
monthly.debit monthly.credit gl_acct unbilled_earned net
monthly.debit monthly.credit gl_acct unbilled_earned disc

```

```

monthly.credit monthly.debit gl_acct unbilled_unearned gross
monthly.debit monthly.credit gl_acct unbilled_unearned net
monthly.debit monthly.credit gl_acct unbilled_unearned disc
monthly.credit monthly.debit
# G/L ID for cancel rate
glid ( id
103
descr Cancel Fees gl_acct billed gross cancel.debit
cancel.credit
gl_acct billed net cancel.debit cancel.credit gl_acct billed
disc cancel.credit cancel.debit gl_acct billed_earned gross
cancel.debit cancel.credit gl_acct billed_earned net
cancel.debit cancel.credit gl_acct billed_earned disc
cancel.credit cancel.debit gl_acct unbilled gross
cancel.debit cancel.credit gl_acct unbilled net
cancel.debit cancel.credit gl_acct unbilled disc
cancel.credit cancel.debit gl_acct unbilled_earned gross
cancel.debit cancel.credit gl_acct unbilled_earned net
cancel.debit cancel.credit gl_acct unbilled_earned disc
cancel.credit cancel.debit
# G/L ID for usage rate
glid ( id
104
descr Dialup Usage Fee
gl_acct billed gross dialup.debit dialup.credit gl_acct
billed net dialup.debit dialup.credit gl_acct billed disc
dialup.credit dialup.debit gl_acct billed_earned gross
dialup.debit dialup.credit gl_acct billed_earned net
dialup.debit dialup.credit gl_acct billed_earned disc
dialup.credit dialup.debit gl_acct unbilled gross
dialup.debit dialup.credit gl_acct unbilled net
dialup.debit dialup.credit gl_acct unbilled disc
dialup.credit dialup.debit gl_acct unbilled_earned gross
dialup.debit dialup.credit gl_acct unbilled_earned net
dialup.debit
dialup.credit gl_acct unbilled_earned disc
dialup.credit dialup.debit

# Your custom G/L IDs below. Start the G/L ID at 1000000. You may
# use the G/L IDs defined previously or delete them and create your
# own. However, if you create your own misc G/L, payment,
# adjustment, dispute/settlement and writeoff G/L ID, you must also
# tell Infranet to assign these events with your newly defined G/L
# IDs by using the reasons codes file or modifying
# fm_act_pol_spec_glid.c.

```

Oracle BRM XML GL Report

This appendix provides a sample Oracle Communications Billing and Revenue Management (Oracle BRM) General Ledger (GL) report.

[Example B-1](#) is a sample of the Oracle BRM XML GL report:

Example B-1 Sample Oracle BRM GL report

```
--<GeneralLedgerReport targetNamespace="http://www.portal.com/schemas/GLSync"
xmlns="http://www.portal.com/schemas/GLSync"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.portal.com/schemas/GLSync brm_gl_data.xsd">
  <SourceSystemID>202</SourceSystemID>
  <ReportID>0.0.0.1-73988-6</ReportID>
  <RevenueType>Unbilled earned</RevenueType>
  <BRM_GL_Segment>.</BRM_GL_Segment>
--<ReportCreatedTime>
  <Year>2007</Year>
  <Month>01</Month>
  <Day>02</Day>
  <Hours>4</Hours>
  <Minutes>14</Minutes>
  <Seconds>51</Seconds>
  </ReportCreatedTime>
--<PeriodStartTime>
  <Year>2007</Year>
  <Month>01</Month>
  <Day>04</Day>
  <Hours>0</Hours>
  <Minutes>0</Minutes>
  <Seconds>0</Seconds>
  </PeriodStartTime>
--<PeriodEndTime>
  <Year>2007</Year>
  <Month>01</Month>
  <Day>05</Day>
  <Hours>0</Hours>
  <Minutes>0</Minutes>
  <Seconds>0</Seconds>
  </PeriodEndTime>
--<RevenueAmounts element="0">
  <ResourceId>84 0</ResourceId>
  <BRM_GL_ID>102</BRM_GL_ID>
--<ARGrossAccount name="monthly.debit">
  <Credit>0</Credit>
  <Debit>0.32</Debit> </ARGrossAccount>
--<ARDiscountAccount name="monthly.credit">
```

```

        <Credit>0</Credit>
        <Debit>0</Debit>
    </ARDi scountAccount>
    -<ARNetAccount name="monthly.debit">
        <Credit>0</Credit>
        <Debit>0.32</Debit>
    </ARNetAccount>
    -<ARTaxAccount name="-- N/A --">
        <Credit>0</Credit>
        <Debit>0</Debit>
    </ARTaxAccount >
    -<OffsetGrossAccount name="monthly.credit">
        <Credit>0.32</Credit>
        <Debit>0</Debit>
    </OffsetGrossAccount>
    -<OffsetDiscountAccount name="monthly.debit">
        <Credit>0</Credit>
        <Debit>0</Debit>
    </OffsetDiscountAccount>
    -<OffsetNetAccount name="monthly.credit">
        <Credit>0.32</Credit>
        <Debit>0</Debit>
    </OffsetNetAccount>
    -<OffsetTaxAccount name="-- N/A --">
        <Credit>0</Credit>
        <Debit>0</Debit>
    </OffsetTaxAccount>
    </RevenueAmounts>
    <RevenueAmounts element="1">
    <ResourceId>84 0</ResourceId>
    <BRM_GL_ID>104</BRM_GL_ID>
    -<ARGrossAccount name="dialup.debit">
        <Credit>0</Credit>
        <Debit>0.10</Debit>
    </ARGrossAccount>
    -<ARDDiscountAccount name="dialup.credit">
        <Credit>0</Credit>
        <Debit>0</Debit>
    </ARDi scountAccount>
    -<ARNetAccount name="dialup.debit">
        <Credit>0</Credit>
        <Debit>0.10</Debit>
    </ARNetAccount>
    -<ARTaxAccount name="-- N/A --">
        <Credit>0</Credit>
        <Debit>0</Debit>
    </ARTaxAccount >
    -<OffsetGrossAccount name="dialup.credit">
        <Credit>0.10</Credit>
        <Debit>0</Debit>
    </OffsetGrossAccount>
    -<OffsetDiscountAccount name="dialup.debit">
        <Credit>0</Credit>
        <Debit>0</Debit>
    </OffsetDiscountAccount>
    -<OffsetNetAccount name="dialup.credit">
        <Credit>0.10</Credit>
        <Debit>0</Debit>
    </OffsetNetAccount>
    <OffsetTaxAccount name="-- N/A --">

```

```
<Credit>0</Credit>  
<Debit>0</Debit>  
</OffsetTaxAccount>  
</RevenueAmounts>  
</GeneralLedgerReport>
```


Oracle BRM GL Report

This appendix provides a sample Oracle Communications Billing and Revenue Management (Oracle BRM) General Ledger (GL) report.

Figure C-1 is a sample of the report created by Oracle BRM:

Figure C-1 Sample Oracle BRM GL Report

General Ledger Data						
Type	unbilled					
Posting Start	Sun Jan 1 00:00:00 2006					
Posting End	Fri Jan 5 00:00:00 2007					
102 (USD)						
DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT	
GROSS monthly debit	39.80	9.95	monthly credit	9.95	39.80	
NET monthly debit	39.80	9.95	monthly credit	9.95	39.80	
DISC monthly credit	0.00	0.00	monthly debit	0.00	0.00	
TAX -- N/A --	0.00	0.00	-- N/A --	0.00	0.00	
104 (USD)						
DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT	
GROSS dialup debit	33.97	7.00	dialup credit	7.00	33.97	
NET dialup debit	33.97	7.00	dialup credit	7.00	33.97	
DISC dialup credit	0.00	0.00	dialup debit	0.00	0.00	
TAX -- N/A --	0.00	0.00	-- N/A --	0.00	0.00	

