Oracle® Communications Billing and Revenue Management Integration Pack for Oracle® E-Business Suite: Revenue Accounting 2.5 - Implementation Guide

Release 2.5

Part No. E17428-02

March 2011

Oracle Communications Billing and Revenue Management Integration Pack for Oracle E-Business Suite:
Revenue Accounting 2.5 - Implementation Guide

Part No. E17428-02

Contents

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

This preface discusses:

- Oracle Application Integration Architecture – Foundation Pack 2.5: Getting Started with the Oracle AIA Foundation Pack and Demo
- Oracle Application Integration Architecture - Foundation Pack 2.5: Core Infrastructure Components Guide
- Oracle Application Integration Architecture - Foundation Pack 2.5: Concepts and Technologies Guide
- Oracle Application Integration Architecture - Foundation Pack 2.5: Integration Developer's Guide
- Oracle Application Integration Architecture Process Integration Packs
- Additional resources

Note: Oracle Application Integration Architecture – Foundation Pack 2.5 guides can be found in My Oracle Support (MOS) Article ID: 955605.1.

Oracle Application Integration Architecture – Foundation Pack 2.5: Getting Started with the Oracle AIA Foundation Pack and Demo

The *Oracle Integration Architecture – Foundation Pack 2.5: Getting Started with Oracle AIA Foundation Pack and Demo* provides information about how Oracle Application Integration Architecture Foundation Pack offers great opportunities to build state-of-the-art service-oriented architecture (SOA) integrations.

This guide is targeted mainly at integration architects who want to extend or adjust prebuilt integrations shipped by Oracle, or who plan to build new SOA-based integrations based on Application Integration Architecture. These concepts and components are demonstrated using the Application Integration Architecture Foundation Pack Demo. This demo is the Application Integration Architecture adaptation of the SOA Order Booking Demo.

Oracle Application Integration Architecture - Foundation Pack 2.5: Core Infrastructure Components Guide

The *Oracle Application Integration Architecture - Foundation Pack 2.5: Core Infrastructure Components Guide* provides conceptual, setup, and usage information for the following core infrastructure components:

- Business Service Repository (BSR).
- Composite Application Validation System (CAVS).
- Error handling and logging.
- The Diagnostics Framework.

Oracle Application Integration Architecture - Foundation Pack 2.5: Concepts and Technologies Guide

The *Oracle Application Integration Architecture - Foundation Pack 2.5: Concepts and Technologies Guide* is a companion volume to the *Oracle Application Integration Architecture - Foundation Pack 2.5: Core Infrastructure Components Guide* and *Oracle Application Integration Architecture - Foundation Pack 2.5: Integration Developer's Guide.* The *Oracle Application Integration Architecture - Foundation Pack 2.5: Concepts and Technologies Guide* provides definitions of fundamental Oracle Application Integration Architecture (Oracle AIA) concepts and discusses:

- Oracle AIA.
- Enterprise business objects (EBOs) and enterprise business messages (EBMs).
- Enterprise business services (EBSs).
- Application business connector services (ABCSs).
- Interaction patterns.
- Extensibility.
- Versioning.
- Business processes.
- Batch processing.
- Infrastructure services.
- Security.

Oracle Application Integration Architecture - Foundation Pack 2.5: Integration Developer's Guide

The *Oracle Application Integration Architecture - Foundation Pack 2.5: Integration Developer's Guide* is a companion volume to *Oracle Application Integration Architecture - Foundation Pack 2.5: Concepts and Technologies Guide* and *Oracle Application Integration Architecture - Foundation Pack 2.5: Core Infrastructure Components Guide.*

The *Oracle Application Integration Architecture - Foundation Pack 2.5: Integration Developer's Guide* discusses how to:

- Create an integration scenario.
- Define business service patterns.
- Design and develop EBSs.
- Design and develop enterprise business flows.
- Design and construct ABCSs.
- Work with message transformation, enrichment, and configuration.
- Develop custom xpath functions.
- Design and construct Java Message Service (JMS) Adapter services.
- Work with enterprise message headers.
- Work with message routing.
- Work with transactions.
- Develop Oracle AIA services to work with CAVS.
- Configure Oracle AIA processes to be eligible for error handling and logging.
- Extend enterprise business objects.

In addition, this guide provides Oracle AIA naming standards.

Oracle Application Integration Architecture Process Integration Packs

A process integration pack (PIP) is a prebuilt set of integrated orchestration flows, application integration logic, and extensible enterprise business objects and services required to manage the state and performance of a defined set of activities or tasks between specific Oracle applications associated with a given process. A PIP provides everything you need to deploy a selected integrated business process area. The PIP product offering is suited to those customers seeking to rapidly implement a discrete business process.

Additional Resources

The table lists the available resources:

Resource	Location
Installation Guide	My Oracle Support
Documentation updates	My Oracle Support
Release Notes	Oracle Technology Network http://www.oracle.com/technology/
Known issues, workarounds, and current list of patches	My Oracle Support

Chapter 1: Understanding the Process Integration for Revenue Management

This chapter provides an overview of the process integration for revenue management and discusses:

- Key benefits.
- Participating applications.
- Revenue Management business process flow.
- Solution assumptions.
- Solution constraints.

Revenue Management Overview

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

Users consider Oracle BRM to be the master for the process integration for revenue management. In Oracle BRM, a GL report generation utility that can be started manually or automatically generates GL data. Oracle BRM produces XML-based files that contain a complete and formatted GL data set based on summary reports. The Oracle Data Integrator (ODI) picks up the reports and transforms them into the 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. ODI 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. 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 integration packs 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 (in the case of E-Business Suite R11.5.10 CU2) or Ledger ID (in the case of E-Business Suite R12.1.1) and Exchange Rate Type mappings for this integration, and it configures other parameters for the other Oracle Communications Process Integration Packs (PIPs).

- The CurrencyCodeMapping domain value map (DVM) maps the Oracle BRM currency codes to their corresponding E-Business Suite values.

Key Benefits

This PIP offers these key benefits:

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

Participating Applications

Participating applications are:

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

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*.

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 comply with 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.

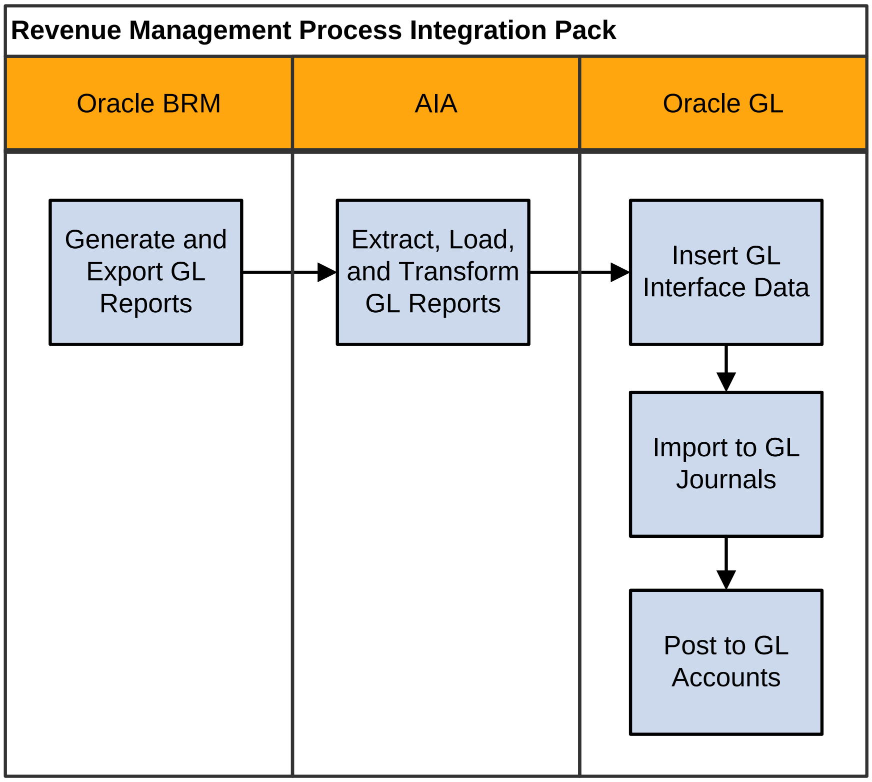
E-Business Suite Release 11.5.10 CU2 uses SOB (Set of Books). Release 12 replaces the SOB with Subledger Accounting (SLA), Ledgers, and Ledger Sets. Important differences exist between SOBs and Ledgers. A SOB is a general ledger with associated subledger books (AR. AP, and so on), and it was self contained with regard to 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 have to 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*.

Revenue Management Business Process Flow

This flowchart illustrates the revenue management business process flow:


Revenue Management Process Integration Pack
Oracle BRM
AIA
Oracle GL
Generate and Export GL Reports
Extract, Load, and Transform GL Reports
Insert GL Interface Data
Import to GL Journals
Post to GL Accounts

Revenue management business process flow

The overall process for revenue management is:

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

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 need to use exact Oracle GL account IDs, then they need to 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 have to manually reverse (N segments × 7 types of reports = M batches) and regenerate the feeds to Oracle GL.

   N could be from tens to hundreds.

6. To address importing nonmonetary journals into Oracle GL (STAT currency), you must create and use a "dummy account" 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 need balanced journals to create journal entries for non-monetary transactions.

   However, for this integration we have to create a balanced Journal Entry. As an example, let us say we are importing 100 free minutes into GL. The Billed Earned report will extract 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 of the accounts should not be used for reporting purposes because it will not provide the actual net amount of 100 free minutes. So, let us say that we create account 7100 for the debit and 7200 (dummy account) for the credit with a classification of expense accounts. The net effect should be 100 minutes, but the transaction will create a DR 7100 for 100 and Credit 7200 for 100. The reports should exclude account 7200 in order to report the 100 minutes. If you need to reduce the amount to 80 free minutes, the corresponding entry would be 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 need to 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 (in the case of E-Business Suite R11.5.10 CU2) or Ledgers (in the case of E-Business Suite R12.1.1) within a single GL instance, you need to 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 (in the case of E-Business Suite R11.5.10 CU2) or a Ledger ID (in the case of 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 need to 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 Ebiz GL does no time zone conversions.

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 will publish data to one SOB ID (in the case of E-Business Suite R11.5.10 CU2) or to one Ledger ID (in the case of E-Business Suite R12.1.1).

   The same SOB ID or Ledger ID can be used by multiple Oracle BRM instances. Thus, 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 folder from which the integration process picks them up, and then stop and restart the agent.

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 will be in the ODIInProcess folder.

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

Chapter 2: Setting Up the Process Integration for Revenue Management

This chapter provides an overview of revenue management integration, lists prerequisites, and discusses how to:

- Perform setup tasks for Oracle E-Business Suite General Ledger (Oracle GL), Oracle Communications Billing and Revenue Management (Oracle BRM), and the integration layer.
- Pick up and transform the GL data using Oracle Data Integrator (ODI).

Revenue Management Overview

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

1. Configure 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 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. In the case of 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.

Prerequisites

You must 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 pack.

For more information, see the *Oracle Application Integration Architecture 2.5: Installation and Upgrade Guide*.

3. Install Oracle BRM GL and complete the initial setup tasks for Oracle BRM GL. 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.
Note: Journal categories are case sensitive and must match the Oracle BRM configuration.

Performing Setup Tasks

This section discusses how to:

- Set up Oracle GL, Oracle BRM, and the integration layer.
- Configure and generate GL data in Oracle BRM.
- Generate data in Oracle BRM.

Setup Tasks for Oracle GL

You must complete these tasks in Oracle GL:

1. Define your Set of Books (SOB) in the case of E-Business Suite R11.5.10 CU2 or ledgers in the case of 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 does not have 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.

Setup Tasks for Oracle BRM

You must 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 (in the case of E-Business Suite R11.5.10 CU2) or ledger ID (in the case of 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 should be the concatenation of the individual Oracle GL segment values separated by the V segment separator string. This segment separator string should 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 assuming that 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 not the same as the Oracle GL account names (including the segment separator). If the Oracle BRM and Oracle GL account names are the same, then you do not need to transform the names using the policy opcode. Also, the segment separator string does not have 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.

Setup Task for the Integration Layer

In 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 (in the case of E-Business Suite R11.5.10 CU2) or ledger ID (in the case of E-Business Suite R12.1.1).

In the case of 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 Application Integration Architecture 2.5: Installation and Upgrade Guide*.

Configuring and Generating GL Data in Oracle BRM

This section discusses how to:

- Configure data in Oracle BRM.
- Generate data in Oracle BRM.

Configuring 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.

Generating 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 will be reported.

In Oracle BRM, you must configure the GL account IDs to correspond with the target accounts in Oracle GL. For information about configuring GL report data in Oracle BRM, see the Oracle BRM documentation.

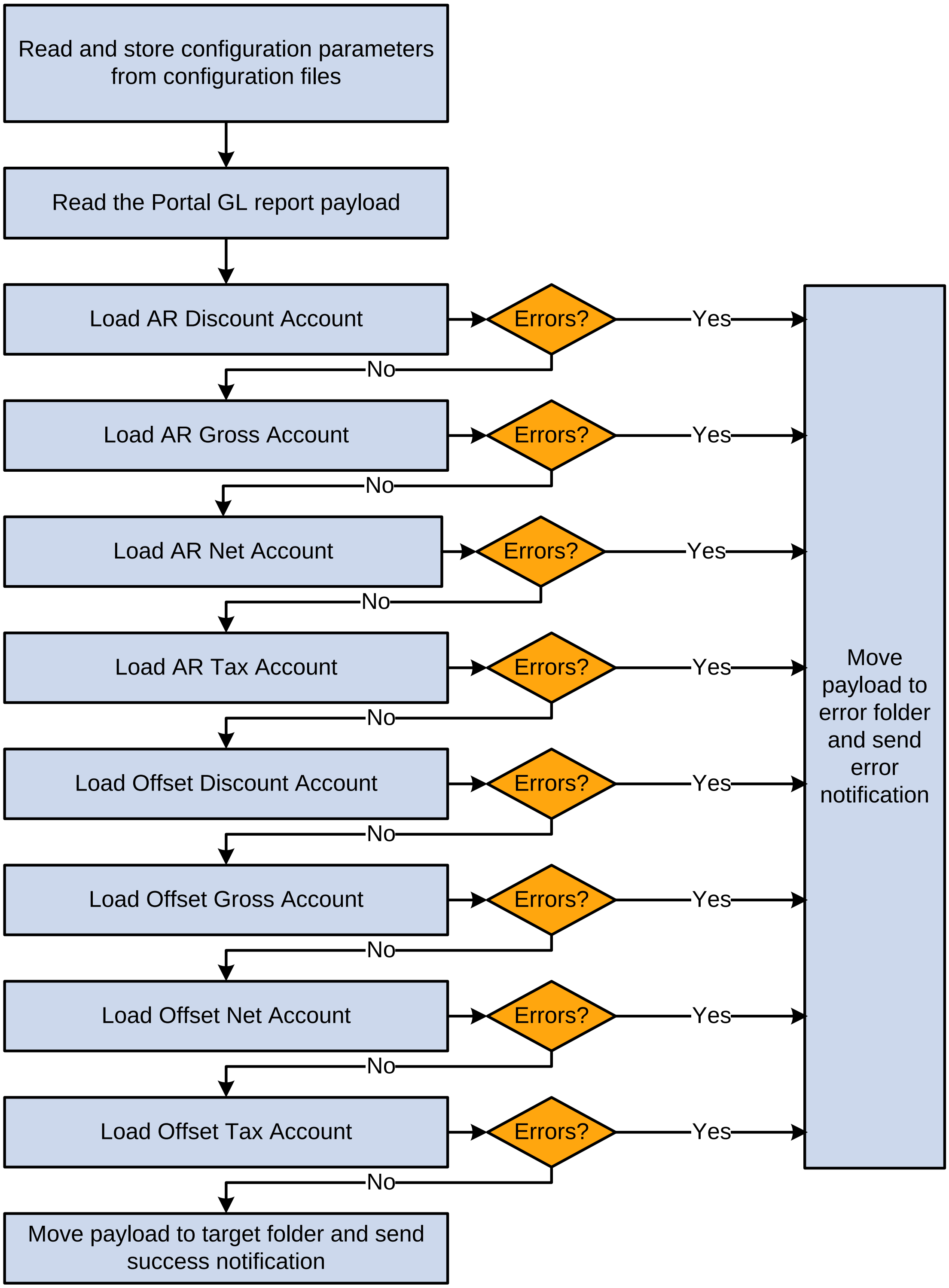
For more information, see *Oracle Communications Billing and Revenue Management (BRM) Documentation*, “BRM Documentation,” Billing Customers, Collecting general ledger data, Loading general ledger configuration data. See also “BRM Documentation,” Billing Customers, Collecting general ledger data, Exporting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

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.

This flowchart illustrates the process:


Read and store configuration parameters from configuration files
Read the Portal GL report payload
Load AR Discount Account
Load AR Gross Account
Load AR Net Account
Load AR Tax Account
Load Offset Discount Account
Load Offset Gross Account
Load Offset Net Account
Load Offset Tax Account
Errors?
Yes
No
Move payload to error folder and send error notification
Move payload to target folder and send success notification

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 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 was successful.

Chapter 3: 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), if required.
- Set up a schedule.
- Configure the process integration for revenue management.
- Run the process integration for revenue management.

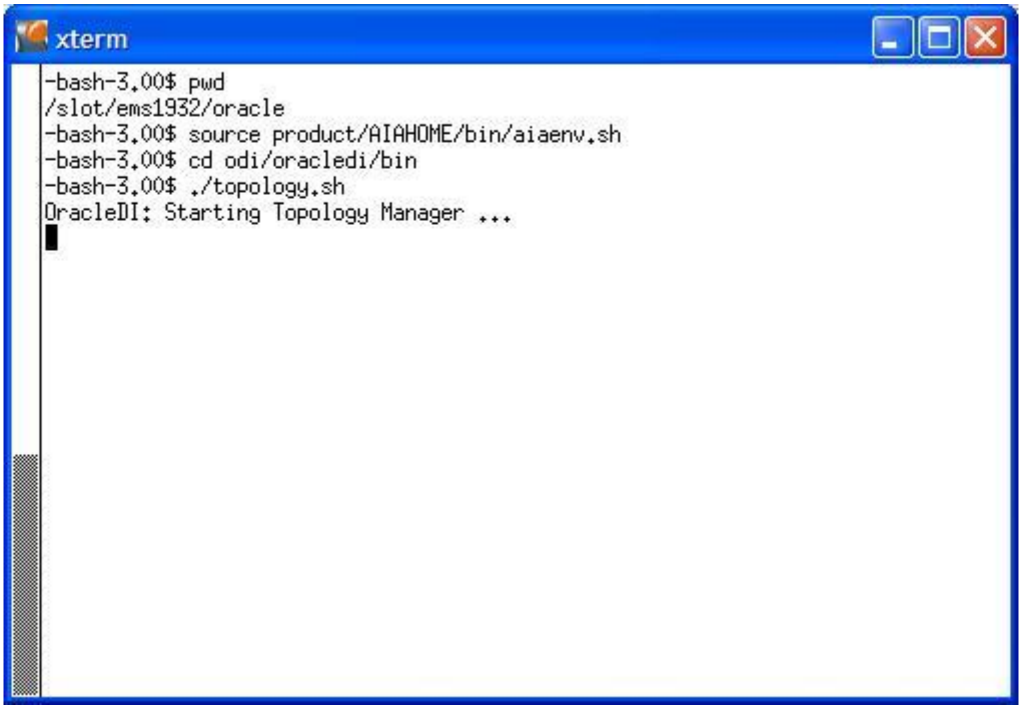
Changing the Oracle GL Database Schema Name

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-install:

1. Open a command prompt.
2. Source the file aiaenv.sh (if UNIX) or aiaenv.bat (if Windows) located in <AIA_HOME>/bin.
3. Navigate to the directory <ODI_HOME>/bin.
4. Open Topology Manager by running the following command:

   If UNIX: ./topology.sh

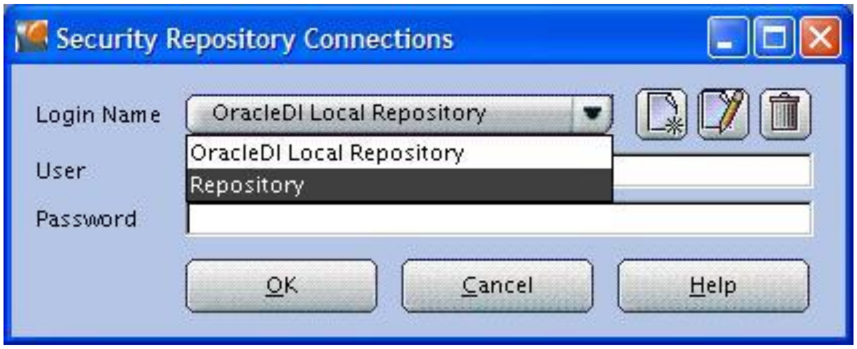
   If Windows: topology.bat



```
-bash-3.00$ pwd
/slot/ems1932/oracle
-bash-3.00$ source product/AIAHOME/bin/aiaenv.sh
-bash-3.00$ cd odi/oracledi/bin
-bash-3.00$ ./topology.sh
OracleDI: Starting Topology Manager ...
```

Example of opening Topology Manager

5. The Security Repository Connections window appears:



Security Repository Connections
Login Name
OracleDI Local Repository
OracleDI Local Repository
Repository
User
Password
OK
Cancel
Help

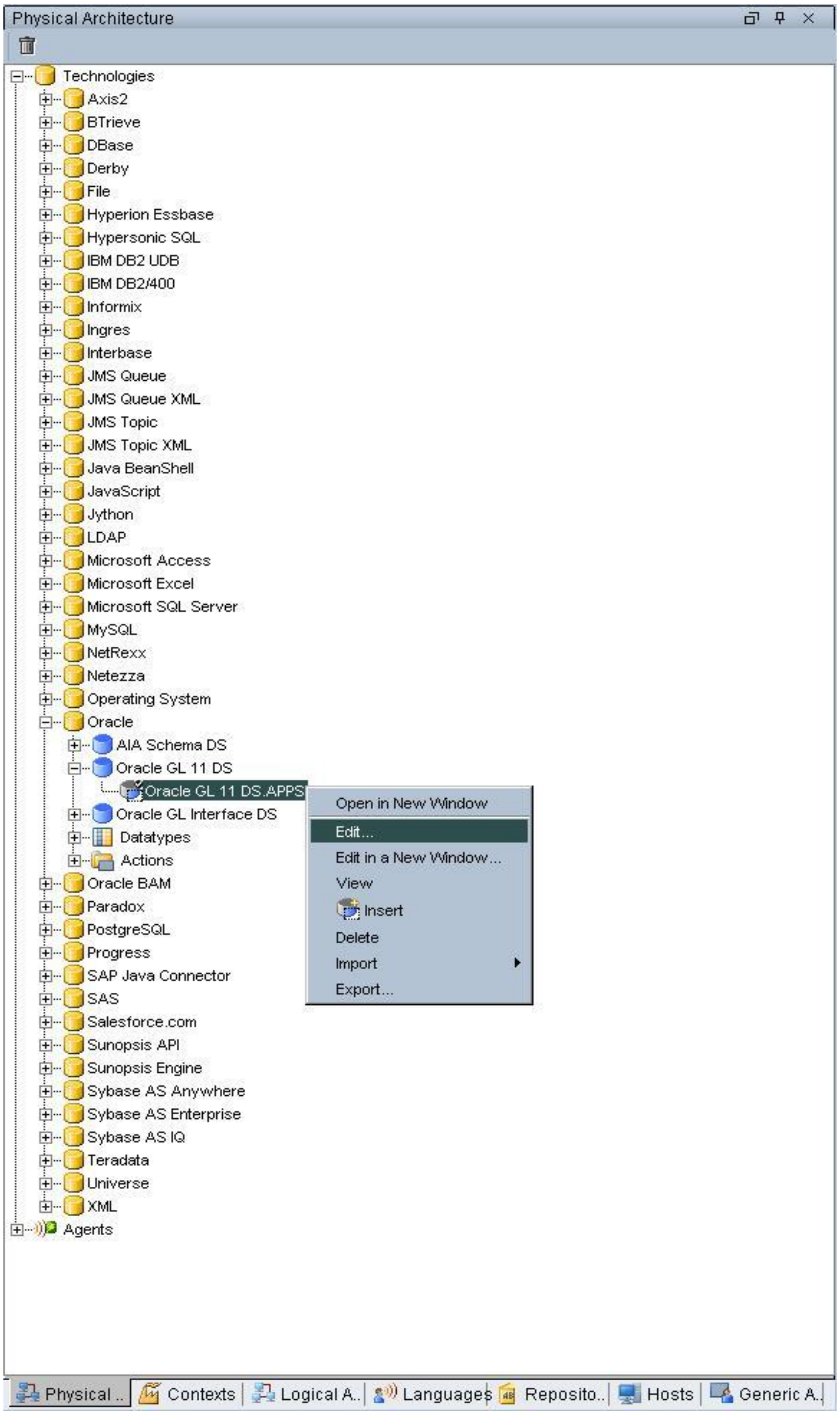
Security Repository Connections window

6. Select Repository in the Login Name field.

7. Enter the user name and password in the identified fields.

8. Click OK.

9. The Topology Manager window appears:



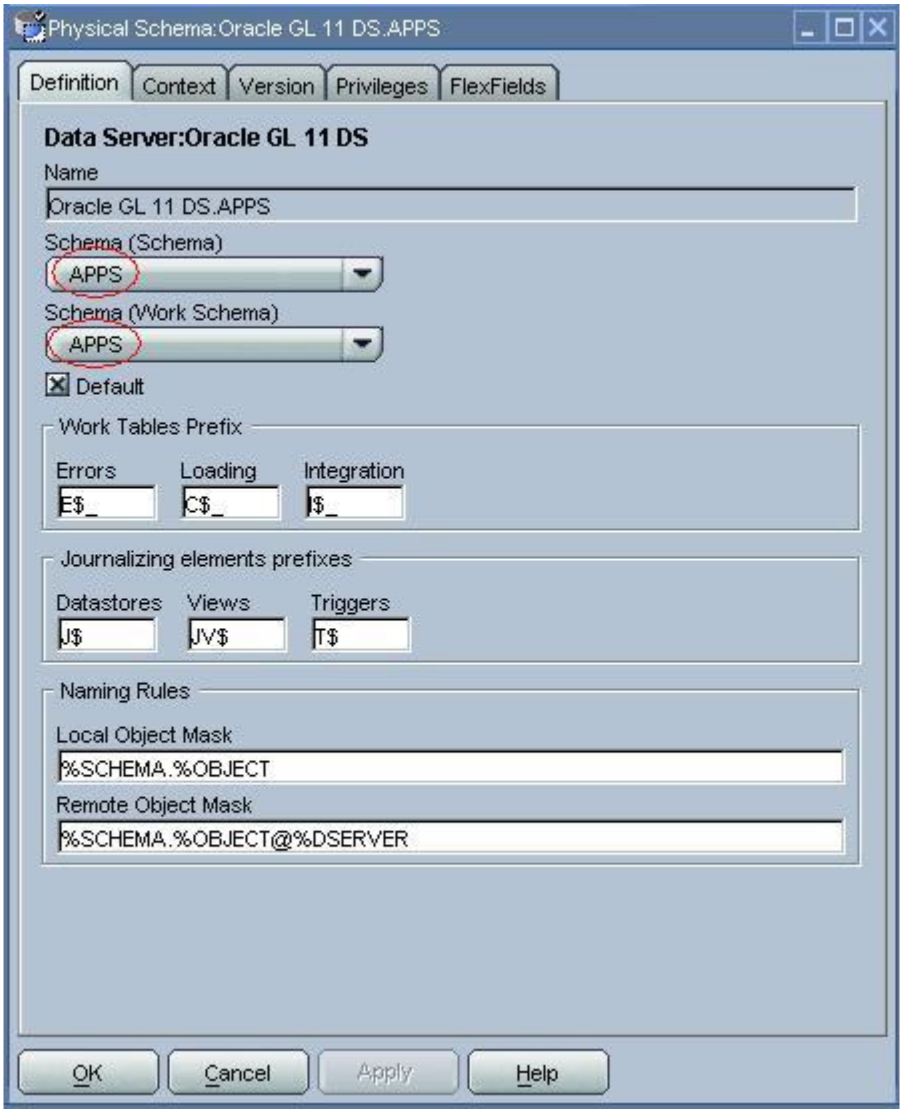
Topology Manager window

10. In the Topology Manager window, navigate to the Physical Schema if you are not there already.

    Expand Technologies > Oracle.

11. If the version of Oracle GL is 11.5.10, expand Oracle GL 11 DS, right-click the child element, and click Edit. (If the version is 12.1.1, see Step 12.)

12. The Physical Schema window opens:



Physical Schema:Oracle GL 11 DS.APPS

Data Server:Oracle GL 11 DS

Name: Oracle GL 11 DS.APPS

Schema (Schema): APPS

Schema (Work Schema): APPS

☒ Default

Work Tables Prefix

Errors: E$_

Loading: C$_

Integration: I$_

Journalizing elements prefixes

Datastores: J$

Views: JV$

Triggers: T$

Naming Rules

Local Object Mask: %SCHEMA.%OBJECT

Remote Object Mask: %SCHEMA.%OBJECT@%DSERVER

Physical Schema window

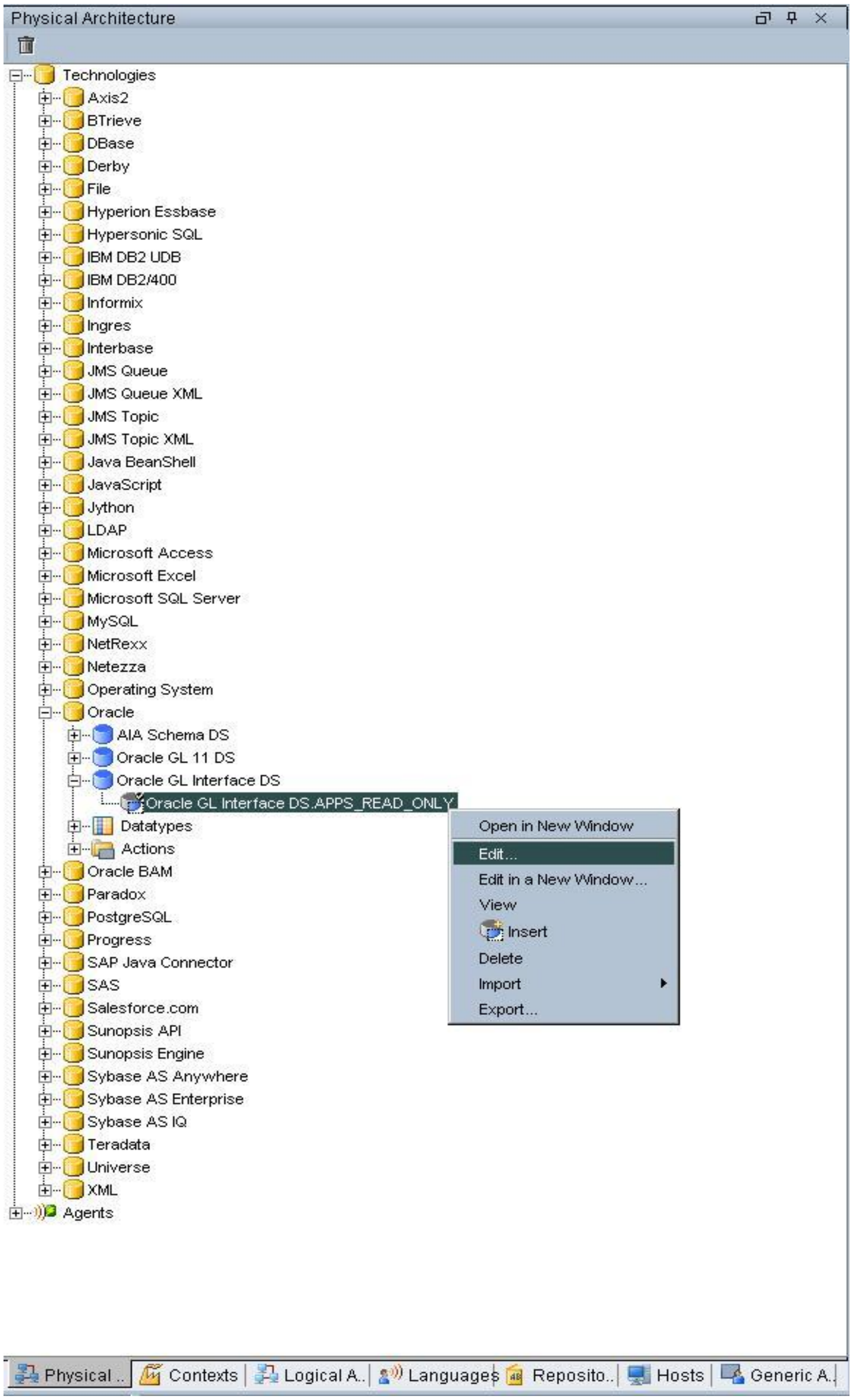
13. Change the values in the Schema (Schema) and Schema (Work Schema) fields to the actual GL database schema name.

Note: Both the Schema and the Work Schema fields should contain the same value.

14. Click the Apply button.

15. Click the OK button.

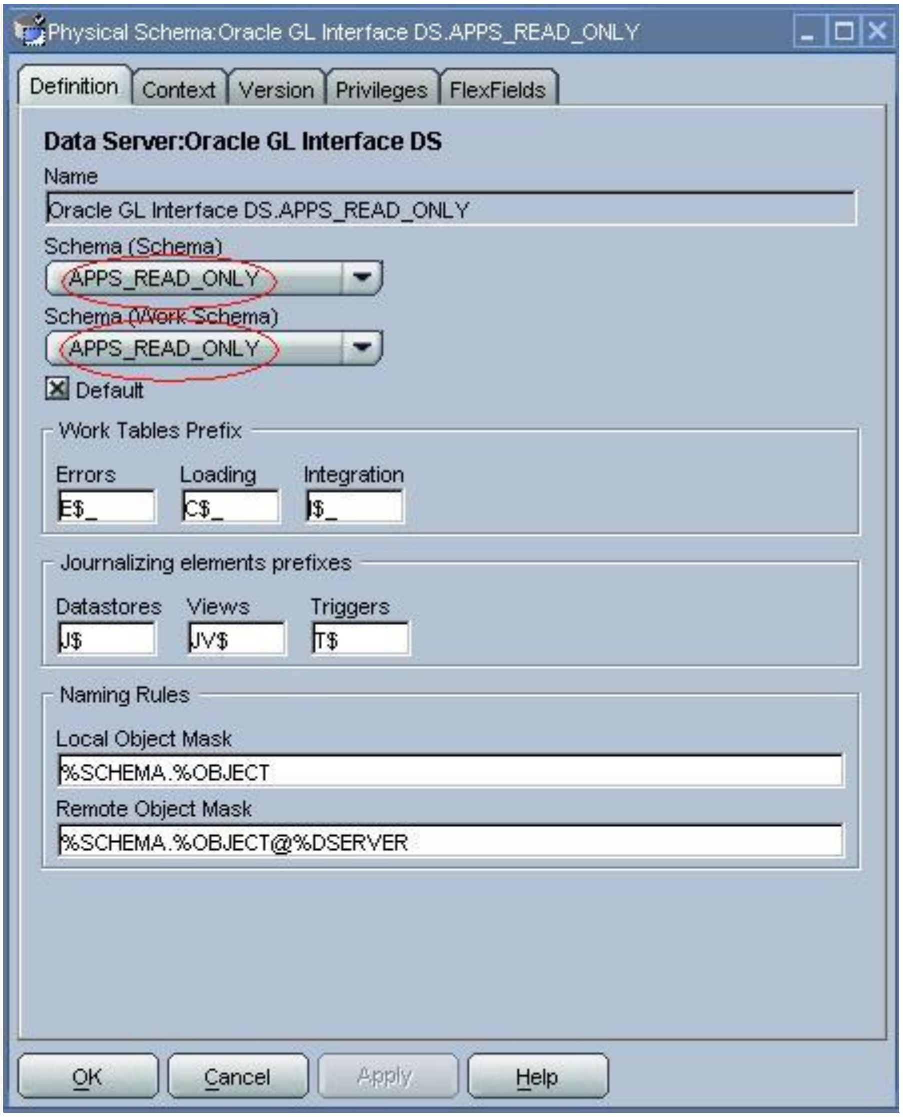
16. If the version of Oracle GL is 12.1.1, expand Oracle GL Interface DS, right-click the child element, and click Edit.



Topology Manager window

17. In the Physical Schema window, change the values of Schema (Schema) and Schema (Work Schema) to the actual GL database schema name.

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



Data Server:Oracle GL Interface DS

Name: Oracle GL Interface DS.APPS_READ_ONLY

Schema (Schema): APPS_READ_ONLY

Schema (Work Schema): APPS_READ_ONLY

☒ Default

Work Tables Prefix

Errors: E$_ Loading: C$_ Integration: I$_

Journalizing elements prefixes

Datastores: J$ Views: JV$ Triggers: T$

Naming Rules

Local Object Mask: %SCHEMA.%OBJECT

Remote Object Mask: %SCHEMA.%OBJECT@%DSERVER

Physical Schema window

18. When done, click Apply.

19. Click OK.

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 2.5: Installation and Upgrade Guide*.

1. Run the following command in the oracledi/bin directory to open the Designer:

   In Windows: designer.bat

   In UNIX: ./designer.sh

2. The Oracle Data Integrator (ODI) Login window appears. Click the New button to create a login. The Work Repository Connection window appears.

3. Enter the following values:

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

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

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

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

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

   f. **Driver List**: Select Oracle JDBC Driver.

   g. **Driver Name**: Enter oracle.jdbc.driver.OracleDriver.

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

   i. **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.

   j. **Default Connection**: Select this check box if you have more than one login name.

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.

5. The Designer window appears.

   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:

   a Context: Select Portal to GL Context.

   b Agent Log: Select RevMgmtAgent.

   c Level: 5.

   d Status: Select the Active option.

   e Execution: Select the On startup option.

8. Click the Execution Cycle tab.
9. Enter or select the following values:

   a Repetition: Select Many times.

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

10. Click OK.

    The scheduler is now configured.

Configuring the Process Integration for Revenue Management

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

Note: Whenever the AIAConfigurationProperties.xml file is updated, you must reload the file for updates to be reflected in the applications or services that use the updated properties. Click the Reload button on the Configuration page in the Oracle AIA Console to reload the file. Alternatively, you can reboot the server to reload.

For more information, see *Oracle Application Integration Architecture - Foundation Pack 2.5: Core Components Guide*, " Loading Oracle AIA Configuration File Updates."

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 BRM instances feeding into 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 make sure that the value of the “SourceSystemID” in the BRM report is one of the values 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 that you want to specify. Do not change the name attribute value. Replace DAILY with the exchange rate type that you want to use.

   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.

   The table lists 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 EBS_01 for 12.1.1 and EBS_02 for 11.5.10 CU2. The default value is EBS_01. Note that the values entered are case sensitive.
ToMailAddress	Specify the 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. Note that the staging/ODIInProcess 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).

   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 will be loaded). If the parameter value is Yes, then the corresponding revenue account type entries are filtered out (they will not be loaded).

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

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

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, if you want 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 Setting Up a Schedule.

Starting the Scheduler Agent

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

In Windows: agentscheduler.bat -NAME=RevMgmtAgent

In UNIX: ./agentscheduler.sh -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.

Stopping the Scheduler Agent

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

In Windows: agentstop.bat

In UNIX: ./agentstop.sh

Appendix A: Sample Oracle BRM GL Configuration File

Here is a sample of the 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 orporation
#    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
#
# Note that you may have different glid files and each glid file may
# have multiple segments. You need to 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 will 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 allows 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. Note that 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
```

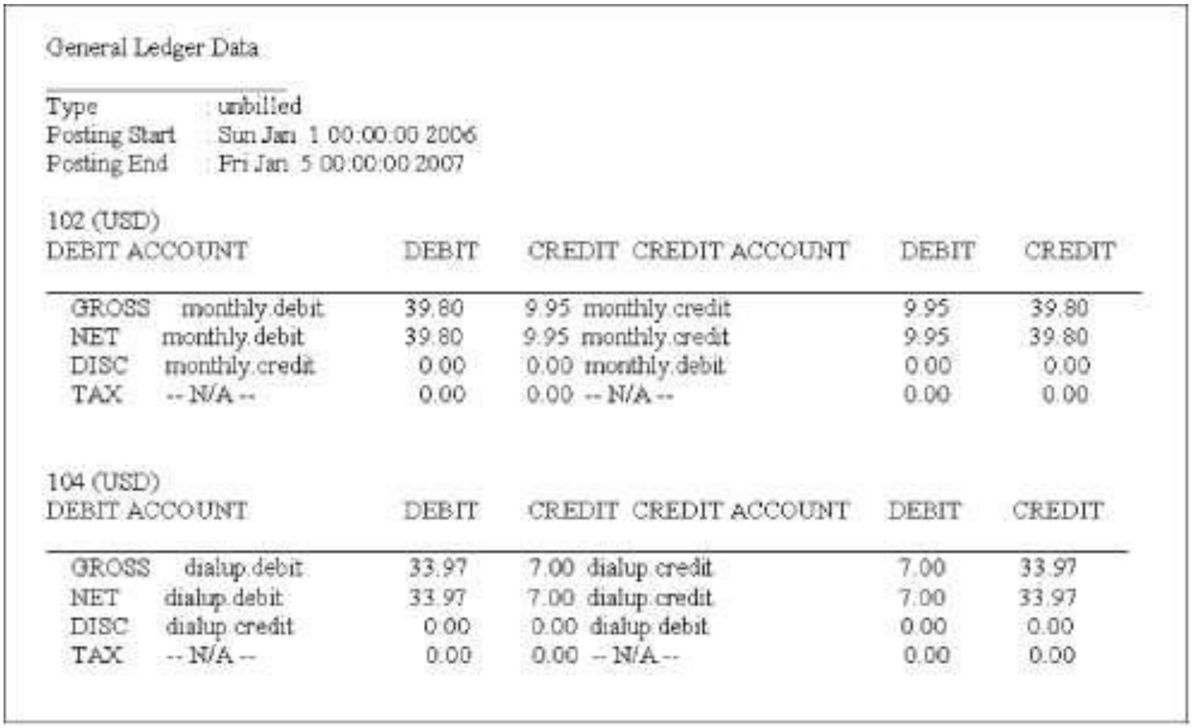
```
#
# This is an optional parameter. If you want the glids to be
# validated against a COA_ID then you must specify this parameter.
# If you do not want validation against a COA then leave it blank or
# do not specify the # parameter.
# If specify this parameter and do not specify any value then it is
# as good as not specifying the parameter.
#
# If a COA_ID value is provided then the program checks if it has
# already 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
#
# Note that 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 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 pull-down 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 should be used.  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 of the Infranet G/L accounts that
#              will be affected by this G/L ID.  For example, an A/R
#              account will appear 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 of the Infranet G/L accounts that
#                 will be affected by this G/L ID.  For example, an
#                 revenue account will appear 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
#  I
#                   |
#  Gross                          |    Asset
#  Net                            |    Asset
#  Tax                            |    Asset
#  Disc               |    Expense or Revenue
#
#
#  For a credit general ledger account
#
#  gl attribute    |    Chart Of Account Type
#  I
#                    |
#  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.
```

Appendix B: Sample Oracle BRM GL Report

Here is a sample of the report created by Oracle BRM:



```
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
```

Sample Oracle BRM GL report

Appendix C: Sample Oracle BRM XML GL Report

Here is a sample of the Oracle BRM XML 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>
  -  <ARDiscountAccount 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>
```

Index