Preface

Preface
Audience
Downloading Oracle Communications Documentation
Documentation Accessibility
Document Revision History

1 About Collecting General Ledger Data
About Collecting General Ledger (G/L) Data
About Implementing General Ledger Reporting

2 List of General Ledger Features
General Ledger Features

3 About General Ledger Revenue Recognition
About Revenue Recognition
Examples of Billed and Unbilled Revenue
Billed and Unbilled Usage Fees
Billed and Unbilled Purchase and Cancellation Fees
Billed Cycle Arrears Fees
Billed and Unbilled Cycle Forward Arrears Fees
Billed and Unbilled Nonrated Events
Billed and Unbilled Cycle Forward Fees
About Unbilled Cycle Forward Fees
About Earned and Unearned Revenue
About Previously Billed Earned Revenue
About Incremental Reports and Cumulative Revenue
About Reversing G/L Entries
About Adjustments and G/L Reporting
How BRM Calculates Earned and Unearned Revenue
Calculating Billed Earned Revenue
Calculating Previously Billed Earned Revenue
Calculating Unearned Revenue
Using Updated G/L Data After an Export .......................................................... 11-2
Correcting Incorrect Data or Replacing Lost Exported Files .......................... 11-3
Restarting a Failed Export ............................................................................. 11-3
Purging G/L Export Data ............................................................................. 11-4

12 General Ledger Utilities

load_pin_config_export_gl ............................................................................. 12-2
load_pin_glchartaccts .................................................................................. 12-3
load_pin_glid ............................................................................................... 12-4
pin_ledger_report ........................................................................................ 12-5
Preface

This guide describes how to use and manage general ledger (G/L) data in Oracle Communications Billing and Revenue Management (BRM).

Audience

This guide is intended for accountants, business planners, operations personnel, and system administrators.

Downloading Oracle Communications Documentation

Product documentation is located on Oracle Technology Network:

http://docs.oracle.com

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

https://edelivery.oracle.com

Documentation Accessibility

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


Access to Oracle Support

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

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

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

Document Revision History

The following table lists the revision history for this book:

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E51012-01</td>
<td>December 2017</td>
<td>Initial release.</td>
</tr>
<tr>
<td>Version</td>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>E51012-02</td>
<td>September 2019</td>
<td>Documentation updates for ECE 12.0 Patch Set 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Updated to include support for Pricing Center and Customer Center.</td>
</tr>
</tbody>
</table>
About Collecting General Ledger Data

This document introduces basic concepts about collecting general ledger (G/L) data in your Oracle Communications Billing and Revenue Management (BRM) system.

If you are ready to collect G/L data, see:

- Creating General Ledger IDs
- Creating a Chart of Accounts
- Running General Ledger (G/L) Reports
- General Ledger Configuration Options
- List of General Ledger Features

About Collecting General Ledger (G/L) Data

The general ledger is the list of accounts that your company uses to track revenue. The general ledger shows how much revenue is recorded for each type of charge; for example, how much revenue is generated by purchase fees or usage fees. Each type of revenue is recorded in a separate general ledger account.

To export general ledger data from the BRM database, you use general ledger IDs (G/L IDs) to identify each type of revenue. You then run a utility to export the data that reports the total amount for each type of revenue. The process is:

1. Create G/L IDs for each type of revenue you need to track. For example, you could track the revenue combined from all types of service usage fees, or you could track usage fees separately for each service. G/L IDs include numbers, such as 102 and a name, such as Monthly fees.

2. When you create charge offers and discount offers, you assign a G/L ID to each balance impact.

3. As charges occur, each event stored in the BRM database includes the balance impact, and a G/L ID. BRM stores revenue data in /journal objects in the BRM database.

4. Every month, you run a utility that creates a general ledger report from the data in the /journal objects. BRM uses the G/L IDs to compile the total amounts for each type of revenue.

Figure 1–1 shows multiple purchase events and monthly fee events compiled into a G/L report.
In addition to generating G/L data about usage fees, purchase fees, and recurring charges, you can collect data about:

- Nonrated events, such as payments and adjustments.
- Tax amounts.

You typically collect G/L revenue monthly, by running the `pin_ledger_report` utility. You can collect and report revenue in the following ways:

- **Post G/L data.** This does not create any reports; it is an internal operation. You post G/L data to maintain the validity of revenue data. Posting G/L data prevents BRM from backdating events that occurred before the posting date; for example, by performing backdated accounts receivable operations.

- **Run G/L reports.** This operation compiles G/L revenue and displays it or loads it into the BRM database. You can generate reports for a variety of options; for example, reports that show billed or unbilled revenue.

  You can run a report without posting the data; for example, if you want to see the revenue status before the normal reporting date. Backdating is still possible if the data is not posted.

Running G/L reports does not export the data from the database. To send data to an external system, you can either export the data by running the `pin_ledger_report` utility, or write a custom application that extracts the data from the `ledger_report` object in the BRM database.

- **Export G/L data.** Exporting G/L data posts the data to the database, runs reports, and exports the reports to XML files. You can then send the XML files to an external accounting system.
To export G/L data, you configure a file that specifies the reports to run, the type of revenue to report, and so on. See "Configuring General Ledger Report Exporting".

**About Implementing General Ledger Reporting**

To implement general ledger reporting:

- Create G/L IDs. See "Creating General Ledger IDs".
- Create a chart of accounts (COA). This list verifies that the G/L accounts defined in BRM are the same as those used in your company. See "Creating a Chart of Accounts".
- Configure G/L exporting. See "Configuring General Ledger Report Exporting".
- Configure various options that specify how G/L data is reported; for example, how G/L reporting recognizes billing dates. See "General Ledger Configuration Options".
- Track different revenue types, such as billed and unbilled revenue, or earned and unearned. See "About General Ledger Revenue Recognition".
- Create G/L IDs for groups of customers, such as customers in a geographic region. See "Reporting Revenue for Groups of Customers".
- Configure how G/L amounts are rounded. See "Rounding General Ledger Data".
This document lists the general ledger features in Oracle Communications Billing and Revenue Management (BRM).

## General Ledger Features

This table lists general ledger features.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learn about collecting general ledger data</td>
<td>About Collecting General Ledger Data</td>
</tr>
<tr>
<td>Learn about how BRM handles revenue recognition</td>
<td>About General Ledger Revenue Recognition</td>
</tr>
<tr>
<td>Configure rounding for general ledger data</td>
<td>Rounding General Ledger Data</td>
</tr>
<tr>
<td>Create a chart of accounts</td>
<td>Creating a Chart of Accounts</td>
</tr>
<tr>
<td>Create general ledger IDs</td>
<td>Creating General Ledger IDs</td>
</tr>
<tr>
<td>Use segments to collect G/L data for groups of customers</td>
<td>Reporting Revenue for Groups of Customers</td>
</tr>
<tr>
<td>Configure business policies for general ledger</td>
<td>General Ledger Configuration Options</td>
</tr>
<tr>
<td></td>
<td>Generating General Ledger Reports Based on the Actual Billing Date</td>
</tr>
<tr>
<td></td>
<td>Segregating Unbilled Revenue by G/L Cycle within a Billing Cycle</td>
</tr>
<tr>
<td></td>
<td>Setting the Number of Paying Accounts per G/L Report</td>
</tr>
<tr>
<td></td>
<td>Disabling Creation of /journal Objects for Noncurrency Balance Impacts</td>
</tr>
<tr>
<td></td>
<td>Disabling General Ledger Collection in BRM</td>
</tr>
<tr>
<td>Run general ledger reports</td>
<td>Running General Ledger (G/L) Reports</td>
</tr>
<tr>
<td>Export general ledger data</td>
<td>Configuring General Ledger Report Exporting</td>
</tr>
<tr>
<td>Troubleshoot general ledger exports</td>
<td>Troubleshooting and Correcting G/L Exports</td>
</tr>
</tbody>
</table>
About General Ledger Revenue Recognition

This document describes how Oracle Communications Billing and Revenue Management (BRM) handles revenue recognition for general ledger reporting. It describes revenue types, such as billed, unbilled, earned, and unearned.

See also:
- Creating General Ledger IDs
- Creating a Chart of Accounts
- Running General Ledger (G/L) Reports
- List of General Ledger Features

About Revenue Recognition

There are two basic types of revenue recognition:

- Billed versus unbilled. You run G/L reports on a regular basis, for example, on the last day of the month. However, customer accounts can be billed on any day of the month. Therefore, at the time the G/L report is run, some accounts have been billed and some have not, so some revenue is recognized as billed and some as unbilled.

- Earned versus unearned. You might run a G/L report that includes revenue from a cycle forward fee that is partially earned and partially unearned. For example, if a customer is charged for one month in advance, if you run a G/L report in the middle of the month, the revenue for the first two weeks has been earned, but the revenue for the last two weeks has not been earned yet, even if it has been billed.

To handle revenue recognition, you need to find out if your company uses immediate revenue recognition or accrual-based revenue recognition:

- Immediate revenue recognition reports all charges as earned as soon as they are applied to a balance. For example, a monthly cycle forward fee is considered earned even if the customer has not used the entire month of service. All revenue is recognized and reported as either billed or unbilled, regardless if it is earned or unearned. You do not need to consider earned or unearned revenue.

- Accrual-based revenue recognition reports all charges as earned based on when the services are rendered. For example, a monthly cycle forward fee is considered only partially earned if the customer has not used the entire month of service. All revenue is recognized and reported as earned or unearned. The revenue is also reported as billed or unbilled. Therefore, revenue can be billed/earned, unbilled/unearned, and so on.
Examples of Billed and Unbilled Revenue

If your company uses immediate revenue recognition, BRM reports all revenue as billed or unbilled.

Billed and Unbilled Usage Fees

In Figure 3–1, the last billing date for the customer was 2/15, and the G/L report was run on 2/28. Usage fees occurring between 2/15 and 2/28 are reported as unbilled.

Billed and Unbilled Purchase and Cancellation Fees

In Figure 3–2, a purchase fee made on 1/20 was billed during the scheduled billing cycle that ran on 2/15. Another purchase fee was made on 2/20. Since the scheduled billing date (2/15) has passed, the revenue for the second purchase fee is recorded as unbilled.

Billed Cycle Arrears Fees

Cycle arrears fees are always billed. In the example illustrated in Figure 3–3, the cycle arrears fee billed on 2/15 is included in the 2/28 report as billed.
Billed and Unbilled Cycle Arrears Fees

In Figure 3–3, the last billing date for the customer was 2/15, and the G/L report was run on 2/28. Cycle arrears fees occurring between 2/15 and 2/28 are reported as unbilled.

Billed and Unbilled Cycle Forward Arrears Fees

In Figure 3–4, the last billing date for the customer was 2/15, and the G/L report was run on 2/28. Cycle forward arrears fees occurring between 2/15 and 2/28 are reported as unbilled.

Billed and Unbilled Nonrated Events

Revenue from nonrated events that are not included in bill items, such as payments and refunds, is always recorded as billed if the event occurs before the G/L report. The billing date has no affect on whether a payment or refund is reported as billed. If a payment or refund event occurs after a G/L reporting date, it is not included in the report.

In Figure 3–5, the payments made on 1/31 and 2/20 are included as billed revenue in the 2/28 G/L report. The payment made on 3/20 is not included in the 2/28 G/L report.
Examples of Billed and Unbilled Revenue

**Figure 3–5  Billed and Unbilled Nonrated Events**

- Billed and Unbilled Cycle Forward Fees
  - Cycle forward fees are always billed except when the cycle forward fee is for a new account. In Figure 3–6, the cycle forward fee billed on 2/28 is reported as billed in the G/L report on 3/15.

**Figure 3–6  Billed and Unbilled Cycle Forward Fees**

- About Unbilled Cycle Forward Fees
  - Cycle forward fees can be unbilled when the fee is for a new account. In Figure 3–7, the cycle forward fee charged when the account is created is not billed until 2/28. Therefore, the G/L report run on 2/15 reports the revenue as unbilled.
About Earned and Unearned Revenue

If your company uses accrual–based revenue recognition, BRM reports revenue as earned and unearned. BRM reports these revenue types:

- Billed earned
- Unbilled earned
- Billed unearned
- Unbilled unearned
- Previously billed earned

Earned revenue is earned at the time that a G/L report is run. Usage, purchase, and cancellation fees are always earned.

Unearned revenue is not earned at the time the G/L report is run. Unearned revenue only applies to revenue from cycle forward fees and cycle forward arrears fees.

Figure 3–8 shows the earned and unearned revenue for a cycle forward fee.

In a cycle forward arrears fee, the revenue is recognized at the start of the cycle, but not billed until the end of the cycle. Figure 3–9 shows earned and unearned revenue for a cycle forward arrears fee.
About Previously Billed Earned Revenue

Previously billed revenue is revenue that was billed in the previous billing cycle but recognized in the current G/L cycle. For example, if a portion of a cycle event, such as a cycle fee, is earned across two G/L cycles, BRM reports the earned portion of this revenue as previously billed earned revenue.

About Incremental Reports and Cumulative Revenue

When you export G/L reports for the following revenue types, the reports include cumulative revenue across reporting periods:

- Unbilled
- Unbilled earned
- Unbilled unearned
- Billed unearned

Part of the total revenue reported for the current period may have been included in the same report for the previous period. The export operation determines the cumulative revenue amounts by calculating the difference between the revenue reported in the current reporting period for a G/L segment and the revenue reported in the previous reporting period for the same G/L segment. (The end date for the previous report is the start date for the current report.) The G/L reports contain the incremental revenue amounts.

When you export G/L reports for nested segments and the revenue type is unbilled, unbilled unearned, unbilled earned, or billed unearned revenue (cumulative revenue),
the segment you report on must be the same from reporting period to reporting period. For example, if a January G/L report calculates unbilled revenue for the .westcoast.california G/L segment, the February G/L report must calculate Unbilled revenue for the .westcoast.california G/L segment. If, instead, the February report calculates unbilled revenue for the .westcoast G/L segment or .westcoast.california.sf G/L segment, the G/L data overlaps and invalid revenue is reported.

**Note:** You can specify overlapping segments for snapshot reports; however, it is not recommended.

### About Reversing G/L Entries

An amount reported as unbilled should be reversed in your company’s general ledger in the period following the one in which it was reported. For example, an entry made for unbilled revenue in the month of January should be reversed if it is billed in February. BRM automatically reverses its general ledger data when it changes from unbilled to billed; however, you might need to reverse the data in your company’s accounting software.

### About Adjustments and G/L Reporting

By default, adjustments made to an open bill item or a pending bill item are always considered billed. The BRM system creates an /event/billing/adjustment/event object and records the adjustment amount as billed in the adjustment item (/item/adjustment). The adjusted amount is reported in the billed G/L report.

In addition, when adjustments are made to a pending bill item, the original amount in the bill item is considered unbilled and is reported in the unbilled G/L report, and the adjustment amount in the adjustment item (item/adjustment) is considered billed and reported in the billed G/L report. When the pending bill items are eventually billed, the billed amount is reported in the billed G/L report.

You can change this behavior to create shadow event objects instead of adjustment event objects. Creating shadow event objects has the following advantages:

- For pending bill item adjustments, both the original amount and the adjustment amount are associated with the same bill item and are recorded in the unbilled G/L report until billing is run.
- Adjustment amounts do not show up in customers’ bills as line items that modify the total due because the events have already been adjusted. You can specify whether shadow event adjustment details are displayed in invoices. See the discussion on customizing information included in invoices in *BRM Configuring and Running Billing*.

**Note:** Shadow events are created for event-level adjustments only. BRM creates shadow events only for events that are adjusted before they are billed.

To specify whether to generate a shadow event or adjustment event, you modify the input list of the PCM_OP_AR_EVENT_ADJUSTMENT opcode. For more information, see the chapter about accounts receivable opcode workflows in *BRM Opcode Guide*. 
How BRM Calculates Earned and Unearned Revenue

BRM uses formulas to calculate earned and unearned revenue. You can use these formulas to determine if revenue is calculated as expected.

Each event in BRM is associated with an item. The status of an item can be pending, open, or closed.

Each item starts in the pending state, gathering events that accrue revenue for the duration of the billing cycle. At the time of billing, the item changes status from pending to open. An open item indicates that payment for the item is due and a bill has been generated. Once payment is received, the status of the item changes to closed.

- Revenue for events associated with a pending item is unbilled revenue.
- Revenue for events associated with an open or closed item is billed revenue.

The PIN_FLD_EFFECTIVE_T field in the /item object holds the date on which an item goes from pending (unbilled) to open (billed) status. This date is used by BRM to determine whether revenue is billed or unbilled.

When BRM creates a cycle event (/event/billing/cycle), the cycle dates for the cycle fee are saved in fields called PIN_FLD_EARNED_START_T and PIN_FLD_EARNED_END_T. The equations in this chapter use the values in those fields. The fields are represented in the equations as shown in Table 3–1:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Equation Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_EARNED_START_T</td>
<td>billing_cycle_start</td>
</tr>
<tr>
<td>PIN_FLD_EARNED_END_T</td>
<td>billing_cycle_end</td>
</tr>
</tbody>
</table>

Calculating Billed Earned Revenue

BRM uses the equation in Figure 3–11 to calculate billed earned revenue:

**Figure 3–11  Billed Earned Revenue Equation**

\[
\text{amount} = \left( \frac{\text{billing}_\text{cycle}_\text{end} - \text{greatest}(\text{gl}_\text{cycle}_\text{end} \text{ or } \text{billing}_\text{cycle}_\text{start})}{\text{billing}_\text{cycle}_\text{end} - \text{billing}_\text{cycle}_\text{start}} \right) \times \text{amount} = \text{earned revenue}
\]

For example, you run the G/L report on the first of every month. A new customer account is created on 7/3, after the G/L report is run on 7/1. Because the new customer’s billing cycle runs from 7/3 to 8/3, the monthly cycle fee covers the period from 7/3 to 8/3. Since G/L is posted on the first of every month, the report run on 8/1 shows the revenue earned for the monthly cycle fees for the period 7/3 to 8/1 in the `billed_earned` category as shown in Figure 3–12.
How BRM Calculates Earned and Unearned Revenue

### Figure 3–12 Calculating Billed Earned Revenue

<table>
<thead>
<tr>
<th>G/L cycle start</th>
<th>Billing cycle start</th>
<th>G/L cycle end</th>
<th>Billing cycle end</th>
</tr>
</thead>
<tbody>
<tr>
<td>7/1</td>
<td>7/3</td>
<td>8/1</td>
<td>8/3</td>
</tr>
</tbody>
</table>

Unearned revenue

If the monthly fee is $100, BRM calculates the earned revenue for the period 7/1 to 8/1 as shown in Figure 3–13:

### Figure 3–13 Sample Earned Revenue Calculation

\[
100 - \left( \frac{2}{31} \times 100 \right) = 93.55
\]

### Calculating Previously Billed Earned Revenue

BRM uses the equation in Figure 3–14 to calculate previously billed revenue:

### Figure 3–14 Previously Billed Revenue Equation

\[
\frac{\text{least(billing_cycle_end or gl_cycle_end)} - \text{greatest(billing_cycle_start or gl_cycle_start)}}{\text{billing_cycle_end - billing_cycle_start}} \times \text{amount}
\]

For example, if a new customer account is created on 7/3 with a bimonthly cycle fee, then the new customer is billed for the next bimonthly cycle fee on 9/3. Since your company requires you to run the G/L report on the first of every month and the cycle fee was created and billed on 7/3, the G/L report run on 8/1 reports the earnings in the `prev_billed_earned` revenue category. This is because the cycle fee was billed before the G/L run on 8/1.

If the bimonthly fee is $120, BRM calculates the previously billed earned revenue as shown in Figure 3–15:
Calculating Unearned Revenue

BRM calculates unearned revenue for cycle fees only. BRM uses the equation in Figure 3–16 to calculate unearned revenue:

Figure 3–16  Unearned Revenue Equation

\[
\left( \frac{\text{billing}_\text{cycle}\_\text{end} - \text{g/l}_\text{cycle}\_\text{end or billing}_\text{cycle}\_\text{start}}{\text{billing}_\text{cycle}\_\text{end} - \text{billing}_\text{cycle}\_\text{start}} \right) \times \text{amount}
\]

For example, if an account is created on 7/3, then the billing cycle will run on the third day of each subsequent month, or on 8/3 as shown in Figure 3–17. If the G/L cycle runs on the first of each month, then the timeline for the billing and G/L cycles is as follows:

Figure 3–17  Calculating Unearned Revenue

For a monthly cycle fee of $19.95, the calculation for unearned revenue is completed as shown in Figure 3–18:
How BRM Calculates Earned and Unearned Revenue

BRM always calculates cumulative unearned revenue. For every successive G/L run, BRM continues to decrease the unearned portion and increase the earned portion. This continues until all revenue is earned by the end date.

If a customer starts a service on 7/3 with a bimonthly cycle fee, then the timeline is as shown in Figure 3–19:

The unearned revenue for August is also reported as unearned in July. The unearned revenue for July and August is calculated as shown in Figure 3–20:

*Figure 3–18 Sample Unearned Revenue Calculation*

\[
\frac{8/3 - \text{greatest}(8/1 \text{ or } 7/3)}{8/3 - 7/3} = \frac{8/3 - 8/1}{8/3 - 7/3} = \frac{2 \text{ days}}{31 \text{ days}}
\]

\[
\frac{2}{31} \times 19.95 = 1.29
\]

*Figure 3–19 Cumulative Unearned Revenue Timeline*

*Figure 3–20 Sample Unearned Revenue Calculations*

**July's calculation**

\[
\frac{9/3 - \text{greatest}(9/1 \text{ or } 7/3)}{9/3 - 7/3} = \frac{9/3 - 9/1}{9/3 - 7/3} = \frac{33 \text{ days}}{62 \text{ days}} + \text{amount}
\]

**August's calculation**

\[
\frac{9/3 - \text{greatest}(9/1 \text{ or } 7/3)}{9/3 - 7/3} = \frac{9/3 - 9/1}{9/3 - 7/3} = \frac{2 \text{ days}}{62 \text{ days}} + \text{amount}
\]
This document describes how Oracle Communications Billing and Revenue Management (BRM) handles rounding in general ledger reporting.

See also:
- About Collecting General Ledger Data
- List of General Ledger Features
- The discussion on rounding in BRM Creating Product Offerings.

**About Rounding and G/L Reports**

To round balance impacts of events for billing and G/L reports, you specify a rounding rule for A/R processes by using PDC. If you use Pricing Center, specify your rounding rules by editing the balance element ID (BEID) configuration file and running the `load_pin_beid` utility. For information, see BRM Setting Up Pricing and Rating.

The balance impacts of events are totaled and rounded separately for billing and G/L reports:
- When a bill is generated, the balance impacts of events in the items are totaled, and then the items are rounded, summed, and added to the bill.
- When a G/L report is generated, the balance impacts of all events with the same G/L ID are totaled in journal entries, and then the journal entries are rounded and posted.

**Rounding G/L Report Data After Billing**

Events that belong to an item can belong to different journal entries. When there are a great number of items, this can cause minor rounding differences between the billing totals and the G/L report totals.

*Figure 4–1* is a simplified example of how billing and G/L totals for the same balance impacts can differ when the balance impacts in an item belong to different journal entries. In this example, the precision is 3 for rating and 2 for A/R, and the mode is round to the nearest:
Rounding G/L Report Data Prior to Billing

When you run billing, this rounding difference is recorded in the bill item, and included in G/L reports. A G/L ID is defined for rounding difference so that the G/L report can be accurately reconciled.

Rounding G/L Report Data Prior to Billing

Because items are not rounded prior to billing, they might have a high precision such as six significant digits. However, journal entries are rounded when G/L reports are run. This can create small rounding discrepancies because rounding differences are not recorded until billing is run. For example, if the sum of all billing items is 100.53009, the pre-billing G/L report will display the rounded amount of 100.53, leaving a difference of 0.00009 undocumented.

<table>
<thead>
<tr>
<th>Item</th>
<th>Journal entry A</th>
<th>Journal entry B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balance impacts after rounding for rating to a precision of 3:</td>
<td>1.111</td>
<td>1.111</td>
</tr>
<tr>
<td></td>
<td>2.131</td>
<td>2.132</td>
</tr>
<tr>
<td></td>
<td>3.141</td>
<td>3.145</td>
</tr>
<tr>
<td></td>
<td>1.223</td>
<td>1.223</td>
</tr>
<tr>
<td></td>
<td>4.112</td>
<td>4.112</td>
</tr>
<tr>
<td>Total</td>
<td>11.723</td>
<td>6.388</td>
</tr>
<tr>
<td>A/R rounded total</td>
<td>11.72</td>
<td>6.39</td>
</tr>
<tr>
<td>Billing total =</td>
<td>11.72</td>
<td>G/L total =</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.73</td>
</tr>
</tbody>
</table>
Creating a Chart of Accounts

This document describes how to create a general ledger (G/L) chart of accounts (COA) for Oracle Communications Billing and Revenue Management (BRM).

See also:
- About Collecting General Ledger Data
- Creating General Ledger IDs

Creating a Chart of Accounts

The chart of accounts (COA) is a list of general ledger accounts that your company uses. By creating a COA in BRM, you can verify that the G/L accounts you enter in the G/L ID definitions are valid. The COA is optional.

If you use multiple G/L segments, you can load multiple COAs and assign a different COA to each segment. See "Reporting Revenue for Groups of Customers".

The COA includes the following fields:
- **coa_id**: the identification number for the chart of accounts. You use this ID when configuring G/L IDs.
- **coa_name**: describes the COA.
- **gl_coa_accts**: lists the G/L accounts that are part of the COA.

Shown below is a sample `pin_glchartaccts` file.

The sample also indicates the type of the account: asset, revenue, expense, or liability. It also indicates that all the accounts are active.

```
gl_chartaccts {
  coa_id    1000
  coa_name  Primary COA
  gl_coa_acct  0 undefined revenue active
  gl_coa_acct  1 undefined asset  active
  gl_coa_acct 49400 prepaid.off revenue active
  gl_coa_acct 49300 monthly.A/R asset  active
  gl_coa_acct 49200 uncollect.A/R asset  active
  gl_coa_acct 49100 uncollect.off revenue active
  gl_coa_acct 40700 cancel.A/R asset  active
  gl_coa_acct 40500 prepaid.A/R asset  active
  gl_coa_acct 40000 purchase.off revenue active
  gl_coa_acct 20160 monthly.off revenue active
  gl_coa_acct 20150 cancel.off revenue active
  gl_coa_acct 11000 purchase.A/R asset  active
  gl_coa_acct 10600 daily.A/R asset  active
```
This example shows a file with one COA. To create multiple COAs, use multiple sets of the `coa_id`, `coa_name`, and `gl_coa_accts` fields.

To create a COA:

1. Go to the `BRM-home/sys/data/pricing/example` directory.
2. Edit and save the `pin_glchartaccts` file. The `pin_glchartaccts` file includes instructions and examples.

   Tip: You need to enter a COA ID when you create G/L IDs. If you create multiple COAs, note the COA IDs.

3. Run the following command to load the COA into the database:

   `load_pin_glchartaccts pin_glchartaccts_file`

   See "load_pin_glchartaccts".
Creating General Ledger IDs

This document describes how to create general ledger IDs (G/L IDs) for Oracle Communications Billing and Revenue Management (BRM).

See also:
- About Collecting General Ledger Data
- Creating a Chart of Accounts
- List of General Ledger Features

Before creating G/L IDs, you need to create a chart of accounts (COA). See "Creating a Chart of Accounts".

Creating General Ledger IDs

To create G/L IDs, you edit the pin_glid file and use the load_pin_glid utility to load the G/L IDs into the BRM database.

Each G/L ID definition contains the following information:
- An identification number.
- A description, such as Monthly fees.
- (Optional) The tax code used by your tax calculation software.
- The G/L account that collects the revenue data, including the revenue type, for example billed, unbilled, and unearned. In addition, you define the following account pair attributes:
  - gross – reports the total of net and discounted revenue.
  - disc – reports the balance impacts of discounted revenue.
  - net – reports the amount of revenue that remains after applying discounts.
  - tax – reports the amount of taxes calculated. This data is used for collecting G/L data based on tax codes.

Usually, you post G/L for gross and discount revenue, or you post for net revenue. For example, if you provide a $5 discount on a $30 cycle forward fee:
- The gross amount is $30
- The discounted amount is $5
- The net amount is $25

This example shows the G/L ID definition for a purchase fee:
Assigning G/L IDs to Nonrated Events

Nonrated events, such as payments and refunds, are included in G/L reports.

To assign G/L IDs to nonrated events:

1. Go to the BRM_home/sysmsgs/data/pricing/example directory.

2. Edit the pin_glid file. The pin_glid file includes instructions and examples.

3. (Optional) Enter the COA to use for this pin_glid file. The default is to not use a COA.

4. (Optional) To create a separate set of G/L IDs for a segment, enter a G/L segment to use for this pin_glid file by using the following syntax:

   `gl_segment root segment.child_segment [no_rollup]`

   where `child_segment` is a nested segment and `[no_rollup]` includes the segment in your reports.

   **Important:** Segment names are case sensitive and cannot have spaces.

5. Save the pin_glid file.

6. Run the following command to load the G/L IDs into the BRM database.

   `load_pin_glid pin_glid_file`

   See "load_pin_glid" for more information.

---

Assigning G/L IDs to Nonrated Events

Nonrated events, such as payments and refunds, are included in G/L reports.

To assign G/L IDs to nonrated events:

1. Go to the BRM_home/sysmsgs/reasoncodes directory.
2. Open the reasons.en_US sample file.
3. Add your G/L IDs, making sure that they match the G/L IDs in the pin_glid file.

**Tip:** You can change the existing G/L IDs, such as 109 (payment), to custom G/L IDs.

5. Run the following command, which loads this change into the /config/map_glid object:

   ```
   load_localized_strings reasons.locale
   ```

### Assigning G/L IDs to Prerated Events

Use the PCM_OP_ACT_POL_SPEC_GLID policy opcode to assign G/L IDs to prerated events or partially rated events. See *BRM Opcode Guide*. 
Reporting Revenue for Groups of Customers

This document describes how to use Oracle Communications Billing and Revenue Management (BRM) general ledger reporting to report revenue for customer segments.

See also:
- About Collecting General Ledger Data
- List of General Ledger Features

About Reporting Revenue for Groups of Customers

By default, BRM reports revenue for all customers. You can use G/L segments to create G/L reports that include data for arbitrary sets of customers, for example, G/L reports for different geographic regions. You specify the segment to run the report on when you run the `pin_ledger_report` utility.

When you use multiple G/L segments, you can use the same G/L IDs for all segments, or create different sets of G/L IDs for different segments.

To collect G/L data about a group of accounts, BRM needs to know which G/L segment applies to each customer account.

- Your BRM system uses a default root segment, which is defined in the Connection Manager `pin.conf` file. By default, all accounts use the root segment.
- To assign G/L segments, you must define the G/L segments by editing policy source code. See the discussion of the `PCM_OP_CUST_POL_PREP_BILLINFO` opcode in `BRM Opcode Guide`.

Creating Nested Segments

You can create nested G/L segments. For example, you can create the following segments:

- northwest
- northwest.washington
- northwest.oregon

This allows you to run reports on all of your Northwest region, including Oregon and Washington, or just on Oregon and Washington specifically.

You can define unlimited levels of nesting.

You can specify if the data for a segment should be included when you run a report for the parent segment. To do so, you use the `no_rollup` entry in the `pin.glid` file to specify that the segment should not be included in the parent report.
Changing the Default G/L Segment

Note: This feature requires that you customize the PCM_OP_CUST_POL_PREP BILLINFO policy opcode.

The default G/L segment is the root segment, symbolized by "." (dot).

To change the default G/L segment:

1. Go to the BRM_home/sys/cm/ directory.
2. Open the Connection Manager (CM) configuration pin.conf file in a text editor.
3. Change the following line:
   - fm_cust_pol gl_segment .
   to this:
   - fm_cust_pol gl_segment segment_name
   where segment_name is the name of the G/L segment you want to use as the default.
4. Save the pin.conf file.
   You do not need to restart the CM to enable this entry.
5. Customize the PCM_OP_CUST_POL_PREP BILLINFO policy opcode. See BRM Opcode Guide.

In this example, .northwest.washington is included in parent reports, but .northwest.oregon is not.

Important: Be sure to specify the parent segment in the report or you will get an error message similar to this:

Root Segment .NSL_REGIONAL is missing. Define the root before the child (.NSL_REGIONAL.BB)

Changing the Default G/L Segment
This document describes how to configure Oracle Communications Billing and Revenue manager (BRM) server options for running and exporting general ledger data.

Topics in this document:

- Generating General Ledger Reports Based on the Actual Billing Date
- Setting the Number of Paying Accounts per G/L Report
- Segregating Unbilled Revenue by G/L Cycle within a Billing Cycle
- Disabling Creation of /journal Objects for Noncurrency Balance Impacts
- Disabling General Ledger Collection in BRM

See also:

- About Collecting General Ledger Data
- List of General Ledger Features
- "About Configuring BRM" in BRM Concepts

Generating General Ledger Reports Based on the Actual Billing Date

Billing can be run on a date past when the billing cycle ended. For example, if a billing cycle ends on February 2, billing might be run on February 3. By default, BRM uses the end of the billing cycle as the date on which G/L reports are based. You can configure BRM to base G/L reports on the date that billing was run instead.

Caution:

- To avoid discrepancies after enabling the use of the actual billing date, make sure, when running the general ledger reports for the first time, that the general ledger reporting period includes both the end date of the bill period and the actual billing date.

- For accounts with skipped billing cycles, the revenue reported as billed in a previous general ledger report period may again be reported as billed in a new general ledger report period. To avoid this discrepancy, before you enable the use of the actual billing date, make sure of the following:
  - There are no subordinate accounts with skipped billing cycles.
  - The general ledger reports for the subordinate accounts with skipped billing cycles have already been generated for a period past the billing date of the paying account.
After enabling the use of the actual billing date, do not disable it.

To use the actual billing date when generating the general ledger reports, you need to configure the BRM database by using SQL. Specifically, you need to create a view on the ITEM_T table and recreate the `item_t_synonym` synonym on the view on the ITEM_T table to enable the use of the actual billing date.

---

**Note:** In a multischema system, run these steps for each schema.

1. Open SQL*Plus:

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

   where:
   - `login` is the login name to use for connecting to the BRM database.
   - `password` is the encrypted password for `login`.
   - `database_alias` is the BRM database alias.

2. Create the `i_item_glseg_eff__id` index on the `gl_segment` and `opened_t` columns of the ITEM_T table:

   ```sql
   SQL> CREATE INDEX i_item_glseg_eff__id ON item_t (gl_segment, opened_t) 
   tablespace tablespace_name storage_clause;
   ``

   where:
   - `tablespace_name` is the tablespace name where you want to create the index.
   - `storage_clause` specifies the storage parameters to use when you create the index; for example, storage (initial 30k next 30k maxextents unlimited pctincrease 0 freelists 2).

3. Create the `item_t_gl_view` view on ITEM_T table with all the columns in the ITEM_T table with the exception of the `effective_t` column of the view pointing to the `opened_t` column of the ITEM_T table.

   For example:

   ```sql
   SQL> CREATE OR REPLACE VIEW item_t_gl_view 
   AS SELECT POID_ID0, ACCOUNT_OBJ_DB, ACCOUNT_OBJ_ID0, ACCOUNT_OBJ_REV, 
   ACCOUNT_OBJ_TYPE, ADJUSTED, ARCHIVE_STATUS, AR_BILLINFO_OBJ_DB, 
   AR_BILLINFO_OBJ_ID0, AR_BILLINFO_OBJ_REV, AR_BILLINFO_OBJ_TYPE, 
   AR_BILL_OBJ_DB, AR_BILL_OBJ_ID0, AR_BILL_OBJ_REV, AR_BILL_OBJ_TYPE, 
   BAL_GRP_OBJ_DB, BAL_GRP_OBJ_ID0, BAL_GRP_OBJ_TYPE, 
   BILLINFO_OBJ_DB, BILLINFO_OBJ_ID0, BILLINFO_OBJ_REV, BILLINFO_OBJ_TYPE, 
   BILL_OBJ_DB, BILL_OBJ_ID0, BILL_OBJ_REV, BILL_OBJ_TYPE, CLOSED_T, CREATED_T, 
   CURRENCY, CURRENCY_OPERATOR, CURRENCY_RATE, CURRENCY_SECONDARY, 
   DELTA_DUE, DISPUTED, DUE_DUE_T, OPENED_T "EFFECTIVE_T", EVENT_POID_LIST, 
   FLAGS, GL_SEGMENT, ITEM_NO, ITEM_TOTAL, MOD_T, NAME, OPENED_T, POID_DB, 
   POID_REV, POID_TYPE, READ_ACCESS, RECVD, SERVICE_OBJ_DB, SERVICE_OBJ_ID0, 
   SERVICE_OBJ_REV, SERVICE_OBJ_TYPE, STATUS, TRANSFERED, WRITEOFF, 
   WRITE_ACCESS FROM item_t;
   ``

4. Create the `item_t_synonym` synonym on the `item_t_gl_view` view on the ITEM_T table:

   ```sql
   SQL> CREATE OR REPLACE SYNONYM item_t_synonym FOR item_t_gl_view;
   ``

5. Recompile the invalid objects:
Segregating Unbilled Revenue by G/L Cycle within a Billing Cycle

By default, the G/L report for unbilled revenue considers the time the journal was created to determine if the revenue was unbilled until the end of the G/L cycle. Because revenue for the same item across multiple ledger cycles is aggregated on the same journal entry, the G/L report shows the unbilled revenue as follows:

- If there is a delay in loading the usage events across multiple ledger reporting cycles, the journal entry for these usage events is created in the new G/L cycle. The unbilled revenue report for the prior G/L cycle does not include these usage events.
- If an item is created in the prior G/L cycle and a usage event for that item occurs in the current G/L cycle, the usage revenue is considered as part of the prior G/L cycle. The unbilled revenue report for the prior G/L cycle includes this usage event.

You can configure BRM to segregate revenue for the same item across multiple G/L cycles by creating a separate journal entry for each of these G/L cycles for this item. This ensures the following:

- If there is a delay in loading the usage events across multiple ledger reporting cycles, the journal entry for these usage events is created in the G/L cycle the event occurred in. The unbilled revenue report for the prior G/L cycle includes these usage events.
- If an item is created in the prior G/L cycle and a usage event for that item occurs in the current G/L cycle, the usage revenue is considered as part of the current G/L cycle. The unbilled revenue report for the prior G/L cycle does not include this usage event.

---

**Note:** Segregating unbilled revenue increases the number of journal objects in the BRM system.

---

BRM makes the journal effective in the G/L cycle in which the event’s endtime falls. In the case of delayed and backdated events that are processed after billing or after G/L posting, the event processing determines the G/L period the journal belongs to.

To segregate the unbilled revenue report for G/L cycle:

1. Go to `BRM_home/sys/data/config`.
2. Create the XML file:
   ```
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```
3. Enable the `SegregateJournalsByGLPeriod` entry.
   ```
   <SegregateJournalsByGLPeriod>enabled</SegregateJournalsByGLPeriod>
   ```
4. Save the file.

5. Load the file:

   pin_bus_params bus_params_billing.xml

6. Stop and restart the CM.

7. (Multischema systems only) Run the pin_multidb script with the -R CONFIG parameter. For more information, see BRM System Administrator’s Guide.

---

**Important:** When the SegregateJournalsByGLPeriod business parameter is enabled:

- BRM assumes that a single G/L reporting cycle is used for all G/L segments in the system.
- The future journals will be created based on the G/L calendar but existing journals will remain same.
- If the existing G/L cycle start dates are modified in the /config/gl_calendar object, future journal creation will be based on the new calendar. However, the previously created journals will remain unchanged and do not reflect the new calendar.

---

**Setting Up G/L Calendar Configuration**

When you segregate unbilled revenue by G/L Cycle within a billing cycle, BRM uses the /config/gl_calendar object to get information related to the G/L reporting cycle. This configuration includes an array of timestamps that indicates the start dates of the G/L cycle. For example, if the G/L reporting cycle starts on the third of every month, the configuration contains 12 timestamps for each year, each indicating the third day of the month.

**Note:** Ensure that the current G/L cycle start date is part of this configuration. The timestamps must match the appropriate time zone of the BRM server.

To load data for the unbilled earned revenue report, manually configure the /config/gl_calendar object by using the PCM_OP_CREATE_OBJ opcode. See "Creating Objects" in BRM Developer’s Guide.

Following is a sample flist to populate monthly G/L cycle start timestamps for three months in the US Pacific time zone.

```
0  PIN_FLD_POID                  POID [0] 0.0.0.1 /config/gl_calendar -1 0
0  PIN_FLD_ACCOUNT_OBJ           POID [0] 0.0.0.1 /account 1 0
0  PIN_FLD_HOSTNAME             STR [0] "<any hostname>"
0  PIN_FLD_NAME                 STR [0] "GL_CALENDAR"
0  PIN_FLD_PROGRAM_NAME         STR [0] "<any program name>"
0  PIN_FLDCALENDAR_DATE         ARRAY [1] allocated 1, used 1
1  PIN_FLD_EFFECTIVE_START_T    TSTAMP [0] (1420099200) 01/01/2015 00:00:00 AM
0  PIN_FLDCALENDAR_DATE         ARRAY [2] allocated 1, used 1
1  PIN_FLD_EFFECTIVE_START_T    TSTAMP [0] (1422777600) 01/02/2015 00:00:00 AM
0  PIN_FLDCALENDAR_DATE         ARRAY [3] allocated 1, used 1
```
Disabling Creation of /journal Objects for Noncurrency Balance Impacts

To improve system performance, you can disable creating /journal objects for noncurrency balance impacts. To disable creating /journal objects for noncurrency balance impacts, run the pin_bus_params utility with the NonCurrencyResourceJournaling business parameter. For information about this utility, see BRM Developer’s Guide.

To disable creation of /journal objects for noncurrency balance impacts:
1. Go to BRM_home/sys/data/config.
2. Create the XML file:
   ```
pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```
3. For the NonCurrencyResourceJournaling entry, do one of the following:
   - To disable creation of all /journal objects for noncurrency balance impacts, change enter **none**:
     ```
     <NonCurrencyResourceJournaling>none</NonCurrencyResourceJournaling>
     ```
   - To create /journal objects for specific noncurrency balance impacts, use this format:
     ```
     ```

   Spaces are not allowed. For example:
   ```
   <NonCurrencyResourceJournaling>+:23466,199990,200000</NonCurrencyResourceJournaling>
   ```
To create /journal objects for all except specific noncurrency balance impacts, use this format:

```xml
```

Spaces are not allowed. For example:

```xml
<NonCurrencyResourceJournaling>-:458745,190000,2000444,77778</NonCurrencyResourceJournaling>
```

**Caution:** If a balance element ID is not valid, the default value all is used.

4. Save the file.
5. Load the file:

   ```bash
   pin_bus_params bus_params_billing.xml
   ```
6. Stop and restart the CM.
7. (Multischema systems only) Run the `pin_multidb` script with the `-R CONFIG` parameter. For more information, see BRM System Administrator's Guide.

---

**Disabling General Ledger Collection in BRM**

By default, General Ledger collection is enabled in BRM.

**Important:** When General Ledger reporting is disabled, /journal objects are not created.

To disable General Ledger collection:

1. Go to `BRM_home/sys/data/config`.
2. Create the XML file:

   ```bash
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```
3. Disable the `GeneralLedgerReporting` entry:

   ```xml
   <GeneralLedgerReporting>disabled</GeneralLedgerReporting>
   ```
4. Save the file.
5. Load the file:

   ```bash
   pin_bus_params bus_params_billing.xml
   ```
6. Stop and restart the CM.
7. (Multischema systems only) Run the `pin_multidb` script with the `-R CONFIG` parameter. For more information, see BRM System Administrator's Guide.
Running General Ledger (G/L) Reports

This document describes how to create Oracle Communications Billing and Revenue Management (BRM) General Ledger (G/L) reports.

Topics in this document:
- About Running the pin_ledger_report Utility
- Creating G/L Reports
- Exporting G/L Reports
- Posting G/L Data
- Displaying a List of Previously-Run Reports

See also:
- About Collecting General Ledger Data
- Configuring General Ledger Report Exporting
- List of General Ledger Features

About Running the pin_ledger_report Utility

Use the pin_ledger_report utility to generate G/L reports, post G/L data, and export G/L data. The pin_ledger_report utility is located in the BRM_home/bin directory. See "pin_ledger_report" for the command line syntax.

Important: For multischema systems, you must run the utility separately against each database schema in your system.

You run the pin_ledger_report utility in different modes. For example, to export G/L data, you run the utility in the export mode:

```
pin_ledger_report -mode export
```

The modes are:
- **run_report**: This mode creates a report. You can choose to post the report or not. See "Creating G/L Reports".
- **export**: This mode generates the report, posts it to the database, and exports it to XML files that can be imported into an external G/L system. See "Exporting G/L Reports" for more information.
post_only: This mode writes G/L report data to the BRM database. It prevents BRM from backdating events that occurred before the posting date. You can also use it to unpost the last posted report.

See "Posting G/L Data".

list_previous: This mode lists previously run G/L reports.

See "Displaying a List of Previously-Run Reports".

Creating G/L Reports

The run_report mode requires the -start parameter. The date is inclusive.

This command runs a G/L report on events that occurred from May 25 to the current date:

```
pin_ledger_report -mode run_report -start 05/25/17
```

For information about the run_report mode syntax, see "pin_ledger_report".

The options for creating G/L reports are:

- Start and end dates. The dates determine which events apply to the reported G/L data. The start date is inclusive, but the end date is noninclusive and defaults to the current date.

```
pin_ledger_report -mode run_report -start 05/25/17 -end 07/25/17
```

- The segment to report. The default is the root segment.

```
pin_ledger_report -mode run_report -start 05/25/17 -segment EastCoastEnterprises
```

- The type of revenue to report. The default is -billed.

```
pin_ledger_report -mode run_report -start 05/25/17 -type unbilled
```

The types are:
- billed
- unbilled
- billed_earned
- unbilled_earned
- billed_unearned
- unbilled_unearned
- prev_billed_earned

Important: When creating a report for billed revenue, you must use SQL to create an index before creating the report, and drop it after running the report. See "Requirement for Creating a Report for Billed Revenue".

- Whether to write the report to an ASCII file. If you do not use the -report parameter, the report is displayed, but not written to a file.

The data is stored in a /ledger_report object. Storing the data in the BRM database can help protect the integrity of your accounting data.
Use the -report parameter with the -test parameter to write the report without creating the ledger_report object.

```
pin_ledger_report -mode run_report -start 05/25/17 -report gl_report
```

- Whether to create a summary report (-summary) or a detailed report (-detail). The default is a summary report.

```
pin_ledger_report -mode run_report -start 05/25/17 -detail
```

- The balance elements to report on; noncurrency (-noncurrency), currency, or both (-all_resource). If you do not use this parameter, only currency balance elements are included.

```
pin_ledger_report -mode run_report -start 05/25/17 -all_resource
```

- Whether to post the G/L data or not. If you do not use the -post parameter, the data is not posted.

The default post date is the current date, but you can use the -end parameter as the post date.

The -post parameter must be the last parameter on the command line.

```
pin_ledger_report -mode run_report -start 05/25/17 -post
```

See "About Collecting General Ledger Data" for information about posting G/L data.

- Whether to run the utility in calculate-only mode (-test). This allows you to display and write the report to a file without creating it in the BRM database.

```
pinLedger_report -mode run_report -start 05/25/17 -test
```

### Requirement for Creating a Report for Billed Revenue

If you are creating a G/L report for billed revenue, you must create an index named `i_item__poid_type` on the ITEM_T table’s POID_TYPE column. This index is required for billed revenue G/L reports only and is not used by other applications. The BRM installer does not create this index, so that it does not impact the performance of other applications.

To prevent the index from impacting performance, create the `i_item__poid_type` index prior to running `pin_ledger_report` and then drop the index after the report is generated. If you need to run `pin_ledger_report` multiple times, create the index just once and drop it after all runs of `pin_ledger_report` have completed.

To create the index, enter the following command:

```
% sqlplus pin/password@databaseAlias
SQL> create index i_item__poid_type on item_t (poid_type) tablespace tablespaceName storageParameters partitionInfo
```

**Note:** You need to specify partitionInfo only if the ITEM_T table is partitioned.
Exporting G/L Reports

The export mode creates a report, posts G/L data, and exports the report to an XML file.

For information about the export mode syntax, see "pin_ledger_report".

Unlike the run_report mode, most of the options for the export mode are defined in a file, not in the command parameters. You must edit and load the pin_config_export_gl.xml file before exporting G/L data. See "Configuring General Ledger Report Exporting".

The options for exporting G/L reports are:

- The segment to report. The default is the root segment.
  
  pinledger_report -mode -export run_report -segment EastCoastEnterprises

  To process reports for two or more G/L segments at the same time, run multiple instances of the utility.

- Regenerate or resend reports. See "Using Updated G/L Data After an Export" and "Correcting Incorrect Data or Replacing Lost Exported Files".

- Restart the export. See "Restarting a Failed Export".

Posting G/L Data

Use the post_only mode to maintain the validity of revenue data. Posting G/L data prevents BRM from backdating events that occurred before the posting date; for example, by performing backdated accounts receivable operations.

pinledger_report -mode post_only -start 05/25/17

For information about the post_only mode syntax, see "pin_ledger_report".

The options for creating G/L reports are:

- The segment to post. The default is the root segment.
  
  pinledger_report -mode post_only -segment EastCoastEnterprises

- The post date. If you don't use this parameter, the current date is used.
  
  pinledger_report -mode post_only -posted 05/25/17

- Unpost a previous post. The last posted date is changed to the date of the previous post. You can only unpost the last previous post. For example, if you post on January 1 and February 1, you can unpost the February 1 post, but not the January 1 post.
  
  pinledger_report -mode post_only -unpost

Displaying a List of Previously-Run Reports

The previously_run mode requires the -start parameter. The date is inclusive.

This command lists the reports that were created from May 25 to the current date:

pinledger_report -mode previously_run -start 05/25/17

For information about the previously_run mode syntax, see "pin_ledger_report".
The options for listing previously run reports are:

- Start and end dates. The dates determine which events apply to the reported G/L data. The start date is inclusive, but the end date is noninclusive and defaults to the current date.

  ```
  pin_ledger_report -mode previously_run -start 05/25/17 -end 07/25/17
  ```

- The segment to report. The default is the root segment.

  ```
  pin_ledger_report -mode previously_run -start 05/25/17 -segment EastCoastEnterprises
  ```

- The type of revenue to report. The default is all types of revenue.

  ```
  pin_ledger_report -mode run_report -start 05/25/17 -type unbilled
  ```

  The types are:
  - billed
  - unbilled
  - billed_earned
  - unbilled_earned
  - billed_unearned
  - unbilled_unearned
  - prev_billed_earned
This document describes how to configure general ledger (G/L) exporting in Oracle Communications Billing and Revenue Management (BRM).

See also:
- About Collecting General Ledger Data
- Running General Ledger (G/L) Reports
- General Ledger Configuration Options
- List of General Ledger Features

About Exported G/L Files

When you export G/L reports, BRM posts, runs, and exports multiple reports, each of them exported to a separate XML file. Exported XML files are saved with the following naming convention:

RevenueType_ReportEndDate_ReportStartDate_ReportId.xml

where:
- **RevenueType** is the abbreviation for the revenue type being reported, for example, **ue** for Unbilled earned and **be** for Billed earned.
- **ReportEndDate** is end date of the current report in **YYYYMMDD** format.
- **ReportStartDate** is the start date for the current report in **YYYYMMDD** format.
- **ReportId** is the ID of the generated report. For details, see "ReportId".

For example:

ue_20070731_20070630_0.0.0.1-123456-10.xml

One XML file is created for each combination of G/L segment and revenue type defined in the export configuration file.

If you configure your G/L reports for the root segment, G/L data for all segments is reported on, except those that roll up to their parent segments. As many XML files will be created as there are report types, and each one will contain data for the entire root segment.

Table 10–1 describes the XML tags in the G/L export files.
### Table 10–1  XML Tags in G/L Reports

<table>
<thead>
<tr>
<th>XML Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceSystemID</td>
<td>The unique ID for the database containing the G/L data.</td>
</tr>
<tr>
<td>ReportId</td>
<td>The unique ID of the G/L report in the BRM system. It comprises the following information, separated by dashes: database number, POID of the /process_audit/export_gl object, and report number. For example, 0.0.0.1-123456-10.</td>
</tr>
<tr>
<td>RevenueType</td>
<td>Revenue type (billed, unbilled, and so on)</td>
</tr>
<tr>
<td>BRM_GL_Segment</td>
<td>The G/L segment being reported on.</td>
</tr>
<tr>
<td>ReportCreatedTime</td>
<td>Report run time.</td>
</tr>
<tr>
<td>PeriodStartTime</td>
<td>The start date and time of the revenue reporting period.</td>
</tr>
<tr>
<td>PeriodStartEnd</td>
<td>The end date and time of the revenue reporting period.</td>
</tr>
<tr>
<td>RevenueAmounts</td>
<td>The revenue reported for a specific G/L segment:</td>
</tr>
<tr>
<td></td>
<td>■ The balance element ID and its associated BRM G/L ID.</td>
</tr>
<tr>
<td></td>
<td>■ A list of revenue accounts and their credit and debit amounts. For example, ARGrossAccount name=&quot;monthly.debit&quot;.</td>
</tr>
</tbody>
</table>

### Configuring G/L Exporting

You configure your G/L reports for export by editing the `pin_config_export_gl.xml` file and loading its contents into the `/config/export_gl` object in the database. The `pin_config_export_gl.xml` file enables you to set up your reporting schedules, revenue types, and other report information. It is located in the `BRM_home/sys/data/config` directory.

Table 10–2 contains the configuration tags you use to set up your G/L reports for export:

### Table 10–2  Configuration Tags for G/L Reports (Export)

<table>
<thead>
<tr>
<th>Use This Tag:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutputDirectory</td>
<td>Specify the file output directory.</td>
</tr>
<tr>
<td>SourceSystemID</td>
<td>Specify a unique ID for the BRM system containing the G/L data. Use this tag if you export from multiple BRM systems and external system needs to identify the source of the data.</td>
</tr>
<tr>
<td>FileNamePrefix</td>
<td>(Optional) Specify a prefix to append to the XML file name. You can use a prefix to identify types of reports.</td>
</tr>
<tr>
<td>ReportInitialStartDate</td>
<td>This value is only used as the report start date the first time a G/L report is exported for the corresponding segment. All subsequent runs of the <code>pin_ledger_report</code> utility for that segment use the end date of the previous G/L report as the report’s start date value. Specify the initial start date for each G/L segment for which data is exported. The format is YYYYMMDD. The initial start date must be specified for the root G/L segment (Segment name=&quot;.&quot;). All G/L segments to be reported use this date. To set a different start date for a specific G/L segment, use the <code>Segment</code> tag and specify a different initial start date for that G/L segment. This date overrides the root segment start date.</td>
</tr>
</tbody>
</table>
### Table 10–2 | Configuration Tags for G/L Reports (Export)

<table>
<thead>
<tr>
<th>Use This Tag:</th>
<th>To Do This:</th>
</tr>
</thead>
</table>
| **Segment**   | Use the `name` attribute to specify a G/L segment to report on. If you specify a nested segment, be sure to include the root segment prefix. For example:  
  ```xml
  <Segment name=.westcoast>
  </Segment>
  ```  
  **Important:** If you specify multiple instances of the same segment, each one must report on a different set of revenue types. |
| **Frequency** | Specify how often G/L reports are generated for the segment:  
  - Daily  
  - Weekly  
  - Monthly  
  - Yearly  
  - Specific Dates  
  **Important:** The frequency for your BRM G/L reports must match the frequency of the reporting schedule in your external G/L system. For example, if the BRM G/L calendar is Monthly, the export frequency should not be Weekly. This creates inaccurate G/L data. |
| **Date**      | If Frequency is Yearly, set the DayOfMonth and Month sub-tags under Date tag to specify the date to generate G/L reports.  
  For Month, specify 01 through 12; for DayOfMonth specify 01 through 31. If the month has fewer days than specified, the last day of the month is used. For example:  
  ```xml
  <Frequency>Yearly</Frequency>
  <Date>
    <DayOfMonth>-29</DayOfMonth>
    <Month>-02-</Month>
  </Date>
  ```  |
| **DayOfMonth**| If Frequency is Monthly, specify the day of the month to generate G/L reports: 01 through 31. If the month has fewer days than specified, the last day of the month is used. |
| **Day**       | If Frequency is Weekly, specify the day of week to generate G/L reports: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday. |
| **Date**      | If Frequency is Specific Dates, add one Month/DayOfMonth pair for each date on which to generate a G/L report.  
  For Month specify 01 through 12; for DayOfMonth specify 01 through 31. If the month has fewer days than specified, the last day of the month is used. For example:  
  ```xml
  <Frequency>Specific Dates</Frequency>
  <Date>
    <DayOfMonth>-29</DayOfMonth>
    <Month>-02-</Month>
  </Date>
  <Date>
    <DayOfMonth>-31</DayOfMonth>
    <Month>-05-</Month>
  </Date>
  ```  |
Configuring G/L Exporting

Sample G/L Report Configuration File

The following example shows the configuration of `pin_config_export_gl.xml` for the root (`.`) segment:

```xml
<GLReportConfiguration>
  <SourceSystemID>Germany</SourceSystemID>
  <OutputDirectory>
    /$PINHOME/GL_output
  </OutputDirectory>
  <FileNamePrefix>test</FileNamePrefix>
  <ReportInitialStartDate>
    <Segment name="">
      <Year>2011</Year>
      <Month>--07--</Month>
    </Segment>
  </ReportInitialStartDate>
</GLReportConfiguration>
```

Table 10–2 (Cont.) Configuration Tags for G/L Reports (Export)

<table>
<thead>
<tr>
<th>Use This Tag:</th>
<th>To Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>RevenueType</td>
<td>Specify the revenue type for the G/L report:</td>
</tr>
<tr>
<td></td>
<td>■ Billed</td>
</tr>
<tr>
<td></td>
<td>■ Billed earned</td>
</tr>
<tr>
<td></td>
<td>■ Prior billed earned (Prior billed earned revenue is also known as Previously billed earned revenue.)</td>
</tr>
<tr>
<td></td>
<td>■ Unbilled</td>
</tr>
<tr>
<td></td>
<td>■ Unbilled unearned</td>
</tr>
<tr>
<td></td>
<td>■ Unbilled earned</td>
</tr>
<tr>
<td></td>
<td>■ Billed unearned</td>
</tr>
<tr>
<td>Important:</td>
<td>If you specify multiple instances of the same G/L segment, each one must report on a different set of revenue types.</td>
</tr>
<tr>
<td>ReportLevel</td>
<td>Specify the type of G/L report to generate: Summary or Detailed. Specify Detailed only when the report is customized using PCM_OP_GL_EXPORT_GL. See BRM Opcode Guide.</td>
</tr>
<tr>
<td>ResourceType</td>
<td>Specify the balance element type to include in the G/L report:</td>
</tr>
<tr>
<td></td>
<td>■ Monetary</td>
</tr>
<tr>
<td></td>
<td>■ Non-monetary</td>
</tr>
<tr>
<td></td>
<td>■ All</td>
</tr>
<tr>
<td>IncludeNonMonetary</td>
<td>If ResourceType is Non-monetary or All, specify the balance element ID of the nonmonetary balance element to include. Use a separate <code>ResourceId</code> tag for each balance element. All other nonmonetary balance elements are excluded from the report. Note: The IncludeNonMonetary and ExcludeNonMonetary tags are mutually exclusive. You can use only one.</td>
</tr>
<tr>
<td>ExcludeNonMonetary</td>
<td>If ResourceType is Non-monetary or All, specify the ID of the nonmonetary balance element to exclude. Use a separate <code>ResourceId</code> tag for each nonmonetary balance element. All other nonmonetary balance elements are included in the report. Note: The IncludeNonMonetary and ExcludeNonMonetary tags are mutually exclusive. You can use only one.</td>
</tr>
</tbody>
</table>
The following example shows a `pin_config_export_gl.xml` file that contains two segments that are exported at different frequencies:

- `eastcoast` is exported monthly.
- `westcoast` is exported on specific dates.

The `eastcoast` segment gets its initial start date from the root segment, but the `westcoast` segment was added at a later date; therefore, it has a different `ReportInitialStartDate` value.
<Segment name=".westcoast">
    <Year>2007</Year>
    <Month>01</Month>
    <Day>01</Day>
</Segment>
</ReportInitialStartDate>

<SegmentList>
    <Segment name=".eastcoast">
        <Frequency>Monthly</Frequency>
        <DayOfMonth>01</DayOfMonth>
        <RevenueTypeList>
            <RevenueType>Billed</RevenueType>
            <RevenueType>Billed earned</RevenueType>
        </RevenueTypeList>
        <ReportLevel>Summary</ReportLevel>
        <ResourceType>All</ResourceType>
        <ExcludeNonMonetary>
            <ResourceID>100000</ResourceID>
            <ResourceID>100001</ResourceID>
            <ResourceID>100002</ResourceID>
            <ResourceID>100003</ResourceID>
        </ExcludeNonMonetary>
    </Segment>

    <Segment name=".westcoast">
        <Frequency>Specific Dates</Frequency>
        <Date>
            <DayOfMonth>29</DayOfMonth>
            <Month>02</Month>
        </Date>
        <Date>
            <DayOfMonth>31</DayOfMonth>
            <Month>05</Month>
        </Date>
        <Date>
            <DayOfMonth>31</DayOfMonth>
            <Month>08</Month>
        </Date>
        <Date>
            <DayOfMonth>30</DayOfMonth>
            <Month>11</Month>
        </Date>
        <RevenueTypeList>
            <RevenueType>Billed</RevenueType>
            <RevenueType>Billed earned</RevenueType>
        </RevenueTypeList>
        <ReportLevel>Detailed</ReportLevel>
        <ResourceType>All</ResourceType>
        <ExcludeNonMonetary>
            <ResourceID>100000</ResourceID>
            <ResourceID>100001</ResourceID>
            <ResourceID>100002</ResourceID>
            <ResourceID>100003</ResourceID>
        </ExcludeNonMonetary>
    </Segment>
</SegmentList>
</GLReportConfiguration>
</BusinessConfiguration>
Validating and Troubleshooting Exported G/L Reports

Setting up G/L Reporting Schedules

The frequency for your BRM G/L reports must match the frequency of the reporting schedule in your external G/L system. For example, if the external G/L calendar is Monthly, the export frequency for your BRM G/L reports should be Monthly. If the two frequencies do not match, inaccurate G/L data will be reported.

You configure the export schedule for your G/L segments by setting the Frequency value for the G/L segment in the pin_config_export_gl.xml file. Each G/L segment you report on can use a different export frequency as long as it matches the reporting frequency for that G/L segment in the external G/L system.

To set up multiple reporting schedules for the same G/L segment, you must specify different revenue types for each report. You cannot specify overlapping revenue types to report on.

For example, to report on billed revenue on a monthly basis and unbilled revenue on a weekly basis for the eastcoast G/L segment, list multiple instances of the eastcoast segment in the SegmentList tag and specify the revenue type for each schedule, like shown in the sample code below:

```
<SegmentList>
  <Segment name=".eastcoast">
    <Frequency>Monthly</Frequency>
    <DayOfMonth>01</DayOfMonth>
    <RevenueTypeList>
      <RevenueType>Billed</RevenueType>
    </RevenueTypeList>
    <ReportLevel>Summary</ReportLevel>
    <ResourceType>Monetary</ResourceType>
  </Segment>
  <Segment name=".eastcoast">
    <Frequency>Weekly</Frequency>
    <Day>Saturday</Day>
    <RevenueTypeList>
      <RevenueType>Unbilled earned</RevenueType>
    </RevenueTypeList>
    <ReportLevel>Summary</ReportLevel>
    <ResourceType>Monetary</ResourceType>
  </Segment>
</SegmentList>
```

Editing the G/L Export Configuration File

To create or modify G/L reports for export, do the following:

1. Open the G/L configuration file (BRM_home/sys/data/config/pin_config_export_gl.xml) in an XML editor or a text editor.
2. Enter the appropriate information into the file. See "Configuring G/L Exporting".
3. Save and close the file.
4. Use this command to load the G/L configuration information into the /config/export_gl object:
   ```
   load_pin_config_export_gl pin_config_export_gl.xml
   ```

Validating and Troubleshooting Exported G/L Reports

Before exporting your G/L reports, validate that your G/L export configuration data was loaded into the database correctly. Do the following:
1. To write loading errors to a log file, run the `load_pin_config_export_gl` utility with the `-d` parameter.

2. Run the `pin_ledger_report` utility in `post_only` mode and verify the `/data/ledger_report` object is updated in the database.

3. Run the `pin_ledger_report` utility in `run_report` mode with the `-report` and `-test` parameters. This will display the G/L report without creating the `/ledger_report` object.
   
   Verify that the revenue types and G/L segments are correct.

4. Run the `pin_ledger_report` utility in `export` mode and verify that the report name, report type, G/L segments, frequency, and revenue types are correct.

5. Verify that the `/process_audit/export_gl` object was created in the database. View the data in this object and verify the data is correct and that the status of the export run is `COMPLETED`. See "Retrieving Audit Data for Exported G/L Reports".
Troubleshooting and Correcting G/L Exports

This document describes how to troubleshoot and correct exported G/L data in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Retrieving Audit Data for Exported G/L Reports
- Using Updated G/L Data After an Export
- Correcting Incorrect Data or Replacing Lost Exported Files
- Restarting a Failed Export
- Purging G/L Export Data

See also:

- About Collecting General Ledger Data
- Configuring General Ledger Report Exporting
- List of General Ledger Features

Retrieving Audit Data for Exported G/L Reports

You can use G/L export audit data to:

- Ensure that reports have been generated for all periods prior to the start date of the next report.
- Generate statistics.
- Troubleshoot reporting errors.

Information about each run of the pin Ledger Report utility is written to the /process_audit/export_gl object in the BRM database. One process audit object is created for each run of the utility.

To view the data in the audit object, read the object with the testnap utility or the Object Browser. See BRM Developer’s Guide.

The /process_audit/export_gl object contains following information:

- The status of the last export run: COMPLETED, IN_PROGRESS, or INCOMPLETE.
  - COMPLETED indicates that the application run has exited normally.
  - IN_PROGRESS indicates that the application is currently running.
Using Updated G/L Data After an Export

When you make changes to the G/L data in the BRM database after you generated the report for that G/L data, run the pin_ledger_report utility in -export mode with the -regenerate parameter. This command does the following:

- Recreates the G/L report in the database.
- Reposts the G/L report.
- Recreates the XML export files.

**Important:** Do not use the -regenerate parameter if the pin_ledger_report utility exits abnormally. Instead, use the -restart parameter. See "Restarting a Failed Export".

If you regenerate a G/L report that was not the last report generated, you must also regenerate all subsequent reports to ensure the data is accurate. For example, if you need to regenerate a report for March, and reports exist for April and May, you should regenerate the April and May reports.

**Important:** A regenerated report does not correct G/L data that was previously imported into the external G/L system. You must reverse the original reports in the external G/L system and then re-import the regenerated data to maintain data integrity.
To regenerate a previously generated G/L report and re-export it to an XML file, run the `pin_ledger_report` utility in `-export` mode with the `-regenerate` parameter:

```
pinLedgerReport -mode export -regenerate reportId
```

where `reportId` is a combination of the database number, the Portal object ID (POID) of the `/process_audit/export_gl` object, and the report number all separated by dashes. For example, if the database number is `0.0.0.1`, the POID is `123456`, and the report number is `4`, type:

```
pinLedgerReport -mode export -regenerate 0.0.0.1-123456-4
```

To find the report ID, look in the previously generated XML report file, if available, or in the export audit data. See "Retrieving Audit Data for Exported G/L Reports".

### Correcting Incorrect Data or Replacing Lost Exported Files

When the G/L data in the BRM database has not changed and the XML export files were either created incorrectly or lost before they could be imported into the external G/L system, run the `pin_ledger_report` utility in `-export` mode with the `-resend` parameter. This command recreates the XML export files. It does not recreate or repost the G/L report in the database.

---

**Important:** Do not use the `-resend` parameter if the `pin_ledger_report` utility exits abnormally. Instead, use the `-restart` parameter. See "Restarting a Failed Export".

---

To export a previously exported G/L report to an XML file, run the `pinLedgerReport` utility in `-export` mode with the `-resend` parameter:

```
pinLedgerReport -mode export -resend report_ID
```

where `report_ID` is a combination of the database number, the POID of the `/process_audit/export_gl` object, and the report number. For example, if the database number is `0.0.0.1`, the POID is `123456`, and the report number is `4`, type:

```
pinLedgerReport -mode export -resend 0.0.0.1-123456-4
```

To find the report number, look in the previously generated XML report file, if available, or in the export audit data. See "Retrieving Audit Data for Exported G/L Reports".

---

**Important:** This command does not regenerate the report contained in the `/audit_object/export_gl` object. For information on regenerating an exported G/L report, see "Using Updated G/L Data After an Export".

---

### Restarting a Failed Export

If the `pinLedgerReport` utility exits abnormally, it attempts to clean up any errors from that run and finishes processing. If it is not successful (the status is not set to COMPLETED after the run), subsequent runs will not work.

To clean up the previous run and finish processing successfully, use the `-restart` parameter:

```
pinLedgerReport -mode export -restart
```
The utility will continue to process the reports from the point at which it left off when it stopped; therefore, no duplicate reports are generated.

**Purging G/L Export Data**

You can purge the following G/L report data:

- Exported XML files on the file system.
  
  Keep the exported XML files for the latest period for each segment and report type until subsequent runs have been made and the XML files have been imported successfully into the external system.

- `/process_audit/export_gl` objects and `/ledger_report` objects in the database.
  
  Keep `/process_audit/export_gl` objects for each G/L segment/report type combination (for example, a billed revenue report for the `.westcoast` G/L segment) until there are at least two newer sets of objects in the database. This ensures you can re-export or regenerate previous G/L reports if necessary.

---

**Important:** Each `/process_audit/export_gl` object contains a reference to a corresponding `/ledger_report` object; therefore, you should purge them at the same time. Before doing so, make sure they are no longer needed.

---

For information on purging data, see "About Purging Data" in *BRM System Administrator’s Guide*. 
This document provides reference information for Oracle Communications Billing and Revenue Management (BRM) General Ledger utilities.

Topics in this document:
- load_pin_config_export_gl
- load_pin_gchartacct
- load_pin_glid
- pin_ledger_report
Use this utility to configure general ledger exporting. See "Configuring G/L Exporting" for more information.

This utility loads configuration data into the BRM database from the BRM_home/sys/data/config/pin_config_export_gl.xml file.

**Location**

BRM_home/bin

**Syntax**

load_pin_config_export_gl pin_config_export_gl.xml [-d] [-v] [-h]

**Parameters**

*pin_config_export_gl.xml*

The file to load.

- **-d**
  Creates a log file for debugging.

- **-v**
  Displays information as the utility runs.

- **-h**
  Displays the utility parameters.

**Results**

The utility notifies you only if it encounters errors.
load_pin_glchartaccts

Use this utility to load a chart of accounts (COA) into the BRM database. See "Creating a Chart of Accounts" for information.

Location

BRM_home/bin

Syntax

load_pin_glchartaccts [-d] [-t] [-v] [-h] pin_glchartaccts_file

Parameters

- **d**
  Creates a log file for debugging.

- **t**
  Runs the utility in test mode. The COA is not loaded into the database.

- **v**
  Displays information as the utility runs.

- **h**
  Displays the utility parameters.

  **pin_glchartaccts_file**
  The file to load.

Results

The **load_pin_glchartaccts** utility notifies you only if it encounters errors.
load_pin_glid

Use this utility to load G/L IDs into the BRM database.
See "About Collecting General Ledger Data".

Location

BRM_home/bin

Syntax

load_pin_glid [-d] [-t] [-v] pin_glid_file |[-h]

Parameters

-d
Creates a log file for debugging.

-t
Runs the load_pin_glid utility in test mode. G/L IDs are not loaded into the database.

-v
Displays information as the utility runs.

-h
Displays the utility parameters.

pin_glid_file
The name and location of the file that defines the G/L IDs.

Results

The load_pin_glid utility notifies you when it successfully creates the G/L IDs.
If the utility displays an error message, look in the utility log file (default.pinlog) to find any errors. The log file is either in the directory from which the utility was started, or in a directory specified in the configuration file.

Important: You must restart the Connection Manager (CM) for any changes to take effect.
pin_ledger_report

Use this utility to collect BRM general ledger (G/L) information, generate G/L reports, and export the reports to XML files.

For more information, see the following:

- About Collecting General Ledger Data
- Running General Ledger (G/L) Reports
- Configuring General Ledger Report Exporting

Location

BRM_home/bin

Syntax Overview

You can use the -mode parameter to run this utility in these modes:

- "run_report Mode Syntax" runs the report. You can choose whether to post the report or not.
- "post_only Mode Syntax" posts the report but does not display the report or write it to a file.
- "list_previous Mode Syntax" lists previously run reports.
- "export Mode Syntax" exports the report to an XML file.

run_report Mode Syntax

See "Creating G/L Reports".

pin_ledger_report -mode run_report
   -start start_date
   [-end end_date]
   [-segment gl_segment]
   [-type billed | unbilled
     billed_earned | billed_unearned |
     unbilled_earned | unbilled_unearned |
     prev_billed_earned]
   [-report file_name]
   [-summary | -detail]
   [-noncurrency | -all_resource]
   [-test]
   [-verbose]
   [-help]
   [-post]

run_report Mode Parameters

-start mm/dd/yy or mm/dd/yyyy
-end mm/dd/yy or mm/dd/yyyy

The start and end dates for revenue to be included in the report. The value you supply for the start date is inclusive, but the value you supply for the end date is noninclusive and defaults to the current date.

-segment gl_segment
Collecting General Ledger Data

pin_ledger_report

Creates a report for a specific G/L segment. See "Reporting Revenue for Groups of Customers". The default is the root segment.

-\texttt{-type billed | unbilled | billed\_earned | billed\_unearned | unbilled\_earned | unbilled\_unearned | prev\_billed\_earned}

Creates a report for the type of revenue you specify. The default is \texttt{billed}. See "About General Ledger Revenue Recognition".

-\texttt{-report file\_name}

Displays the output of the utility. Use the \texttt{-report} parameter with the \texttt{-test} parameter to display the report without creating the \texttt{/ledger\_report} object.

-\texttt{-summary | detail}

Creates a detailed report or a summary report. By default, summary customer G/L reports are created.

-\texttt{-noncurrency | -all\_resource}

Reports G/L data for noncurrency balance elements only or for both currency and noncurrency balance elements. If you do not use either of these parameters, the utility reports currency only.

-\texttt{-test}

Runs the utility in calculation mode only.

-\texttt{-verbose}

Displays information as the utility runs.

-\texttt{-help}

Displays the utility parameters.

-\texttt{-post}

Posts the report using the value in the \texttt{-end\_date} parameter as the post date. If used, this must be the last parameter in the command line.

\textbf{post\_only Mode Syntax}

See "Posting G/L Data".

\begin{verbatim}
pin_ledger_report \texttt{-mode post\_only}
  [-segment \texttt{gl\_segment}]
  [-verbose]
  [-help]
  [-posted \texttt{posted\_date} | -unpost]
\end{verbatim}

\textbf{post\_only Mode Parameters}

-\texttt{-segment \texttt{gl\_segment}}

Posts the G/L report for a specific G/L segment. See "Reporting Revenue for Groups of Customers".

-\texttt{-verbose}

Displays information as the utility runs.

-\texttt{-help}

Displays the utility parameters.

-\texttt{-posted \texttt{posted\_date} | -unpost}
Posts or unposts the specified ledger report. You specify the report by entering the
post date. To display a list of previously run reports, see "list_previous Mode Syntax".

list_previous Mode Syntax
See "Displaying a List of Previously-Run Reports".

pin_ledger_report -mode list_previous
 -start start_date
 [-end end_date]
 [-segment gl_segment]
 [-type billed | unbilled
   billed Earned | billed_unearned |
   unbilled Earned | unbilled_unearned |
   prev_billed Earned]
 [-verbose]
 [-help]

list_previous Mode Parameters
- start start_date
- end end_date
The start and end times for listing the reports. The default for the end date is the
current date.

- segment gl_segment
Shows the report for a specific G/L segment. The default is to list reports for all
segments. See "Reporting Revenue for Groups of Customers".

- type billed | unbilled | billed Earned | billed_unearned | unbilled Earned | unbilled_unearned | prev_billed Earned
Lists reports for a specified type of revenue. By default, the utility lists all types. See
"About General Ledger Revenue Recognition".

- verbose
Displays information as the utility runs.

- help
Displays the utility parameters.

export Mode Syntax
See "Exporting G/L Reports".

pin_ledger_report -mode export
 [-segment gl_segment]
 [-resend report_ID]
 [-regenerate report_ID]
 [-restart]
 [-verbose]
 [-help]

export Mode Parameters
- segment gl_segment
Exports XML reports for the G/L segment specified. The default is to generate and
export reports for all segments. See "Reporting Revenue for Groups of Customers".
Tip: To process reports for two or more G/L segments at the same time, run multiple instances of the utility.

-resend report_ID
Re-creates the XML files for the specified reports without re-creating the /ledger_report object. See "Correcting Incorrect Data or Replacing Lost Exported Files".

-regenerate report_ID
Re-creates the /ledger_report object and the XML report. See "Using Updated G/L Data After an Export".

-restart
Cleans up a previous abnormal run and finishes processing successfully. See "Restarting a Failed Export".

-verbose
Displays information as the utility runs.

-help
Displays the utility parameters.

Results
If the utility displays an error message, check the utility log file to determine the source of the error and correct it.

If the utility encounters an error in -export mode, either rerun the pin_ledger_report utility in -export mode or allow the next scheduled run of the utility to continue processing the remaining reports. The pin_ledger_report utility will correctly process the remaining reports.

If the utility encounters an error during -export mode when the -resend or -regenerate parameter was used, run the utility again after correcting the errors based on the log file messages. If some of the reports were successfully re-exported or regenerated, run the utility again for the failed reports.